INTERACTIVE COMPUTER BASED MUSIC SYSTEMS

A thesis presented for the degree of
Doctor of Philosophy in Electrical Engineering
in the University of Canterbury,
Christchurch, New Zealand.

by

W.H. TUCKER B.E. (Hons)

1977

To Pauline

With Thanks

Not that the story need be long, but it will take a long while to make it short.

Henry David Thoreau

ABSTRACT

The application of digital computing techniques to music is considered herein, with particular emphasis on interactive systems. Three distinct topics are discussed: computer aids for musicians, sound synthesis, and pitch trajectory estimation. A comprehensive review of the literature pertaining to each topic is presented.

An interactive "Piano Typewriter" system which permits music played on an electronic organ to be recorded for subsequent playback, display and editing is described. Two notations are incorporated - MOD (a positional notation) and TRAD (a subset of conventional notation). Particular attention is paid to the transcription of keyboard performance information for TRAD display, and its subsequent editing. Examples which illustrate the present state of the system are presented.

A digital synthesiser which is implemented in dedicated hardware and which permits a wide range of sound timbres to be generated and performed interactively under computer control is described.

Techniques for pitch trajectory estimation are reviewed in detail, and are assessed with respect to their suitability for wide-band musical signals. Numerical transform techniques for evaluating correlations are also reviewed. A new pitch estimation algorithm which operates in the time-domain by recognising recurring "features" of the signal waveform is described and is related analytically to autocorrelation analysis. The performance of the new algorithm is compared with that of the Gold and Rabiner parallel processing algorithm, and it is concluded that the former is more suitable for most music and speech signals.

ACKNOWLEDGEMENTS

I am especially grateful to my supervisor, Prof. R.H.T. Bates, for his continual guidance, enthusiasm and encouragement. I also wish to thank my associate supervisor, W.K. Kennedy, for his invaluable assistance and patience.

I am grateful to my colleagues M.R. Lamb, Susan D. Frykberg, R.J. Howarth and R.G. Vaughan for their cooperation and assistance. I also wish to thank J.E. Cousins and D.F. Sell of the Music Department. The assistance of Dr J.H. Andreae, N. Gray and F.M. Cady of the Electrical Engineering Department, and former postgraduate students Dr R.B. Jordan and A.B. Robson is gratefully acknowledged. Special thanks are due to Dr I.H. Witten of the Department of Electrical Engineering Science of the University of Essex, England, for his comments on a draft of Chapter 7.

The financial contribution of the New Zealand University Grants Committee towards the establishment of the Electrical Engineering Department's Hybrid Computer Laboratory is acknowledged.

I wish to thank the New Zealand Post Office for their cooperation in releasing me to carry out the work described herein, and for their financial contribution to the preparation of this thesis.

Finally, I owe a special debt of gratitude to Mrs E.M. Bönisch, my parents, and my wife and daughters for their cooperation, encouragement and assistance.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

CHAPTER Page

GLOSSARY

Unless indicated otherwise, symbols used in this thesis have the meanings given below.

ADC	Analogue to digital converter
BDI	Binary data interface
BNF	Backus Naur form
CCD	Charge coupled device
C_L	Clip threshold
CPU	Central processor unit
CTμL	Complementary transistor micro logic
$c(\tau)$	Cepstrum: $c(\tau) \triangleq \mathcal{F}[\log\lvert\mathcal{F}[s(t)]\rvert^2]$
DAC	Digital to analogue converter
DAM	Digital to analogue multiplier
dB	decibel
DFT	Discrete Fourier transform
DMA	Direct memory access
DMAC	Direct memory access channel
DWT	Discrete Walsh transform (see $\mathcal{W}[\cdot]$)
ECG	Electrocardiogram
ECL	Emitter-coupled logic
EEC	Electroencephalogram
EMG	Electromyogram
ETS	Equally tempered scale
$\mathcal{F}[\cdot]$	Forward Fourier transform:

$$\mathcal{F}[s(t)] = S(\omega) = \int_{-\infty}^{\infty} s(t) \exp[-j\omega t]\, dt$$

$\mathcal{F}^{-1}[\cdot]$ Inverse Fourier transform

$$\mathcal{F}^{-1}[S(\omega)] = s(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} S(\omega)\exp[j\omega t]\,d\omega$$

F_o Pitch frequency: $F_o = 1/T$

f_s Sampling frequency: $f_s = 1/\Delta_t$

FFT Fast Fourier transform

FNT Fermat Number transform (see Section 8.3)

FWT Fast Walsh transform

gcd greatest common divisor

$\mathcal{H}[\cdot]$ Hilbert transform:

$$\mathcal{H}[f(x)] = f(x) * \frac{-1}{\pi x}$$

I^+ The set of all positive integers

I/O Input-output

$\{L\}$ Logical (dyadic) autocorrelation function:

$$\{L_j\} = \{f_j\} \circledcirc \{f_j\}$$

min(a,b) Minimisation function: min(a,b) = a if a < b

= b if b < a

MNT Mersenne number transform (see Section 8.3)

mod F Modulo F

MOS Metal oxide semiconductor

MOD MODern music notation (see Section 3.7)

MUX Multiplexer

$\{P_F\}$ Discrete Fourier power spectrum:

$$\{P_F\} = \{|\mathcal{F}[s(t)]|^2\}$$

$\{P_W\}$ Discrete Walsh power spectrum:

$$\{P_W\} = \{(\mathcal{W}[s(t)])^2\}$$

PCM Pulse code modulation

RT — Rader transform (see Section 8.3)

RTL — Resistor-transistor logic

S,A,T,B — Soprano, alto, tenor, bass

SAW — Surface acoustic wave

$\mathrm{sgn}(x)$ — $= 1$ if $x \geq 0$; $= -1$ if $x < 0$

T — Period or pitch period

T_{A-L} — Transformation from arithmetic to logical (dyadic) autocorrelation function

T_{L-A} — Transformation from logical (dyadic) to arithmetic autocorrelation function

TRAD — Conventional music notation

TTL — Transistor-transistor logic

VCO — Voltage controlled oscillator

$\mathcal{W}[\cdot]$ — Discrete Walsh transform:

$$\mathcal{W}[\{f_j\}] = \{F_k\} \quad \text{where} \quad F_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \, \mathrm{wal}(k,j)$$

and where the sequences {f} and {F} are real and of length N

WFTA — Winograd Fourier transform algorithm

XOR — Exclusive Or (see $\circledast$)

z-transform $\quad \{r_k\} \overset{zT}{\leftrightarrow} R(z)$ where

$$R(z) = \sum_{k=-\infty}^{\infty} r(k\Delta_t) \, z^{-k}$$

$\delta(\cdot)$ — Dirac delta function

Δ_t — Sample interval: $\Delta_t = 1/f_s$

$\phi(\tau)$ (Arithmetic) autocorrelation function:

$$\phi(\tau) = \int_{-\infty}^{\infty} s(t)\, s(t+\tau)\, dt$$

where $s(t)$ is real and of finite energy

σ^2 variance

$\psi(t)$ the analytic signal:

$\psi(t) = s(t) + j\,\hat{s}(t)$ where $\hat{s}(t) = \mathcal{H}[s(t)]$

τ delay parameter

$::=$ "is replaced by"

$\triangleq$ "is defined as"

$\equiv$ "is identical to"

$\simeq$ "is approximately equal to"

$<$ ($\leq$) "is less than (or equal to)"

$>$ ($\geq$) "is greater than (or equal to)"

$\{\cdot\}$ denotes a set

ε is a member of (a set)

'n octal designation: $'n = n_8$

$\underset{\sim}{R}$ denotes a vector, e.g. $\underset{\sim}{R} = \{r_k\}$

$[t_1, t_2]$ denotes the interval $t_1 \leq t \leq t_2$

$*$ Convolution operator:

$$s(t) * h(t) = \int_{-\infty}^{\infty} s(\tau)\, h(t-\tau)\, d\tau$$

$\circledast$ Logical (dyadic) convolution operator:

$$\{f_s\} \circledast \{g_s\} = \{h_s\} \text{ where } h_s = \frac{1}{N} \sum_{r=0}^{N-1} f_r \, g_{s \oplus r}$$

and where the sequences {f}, {g} and {h} are each of length $N \in I^+$

$\oplus$ Addition modulo 2 (i.e. exclusive or):

$$0 \oplus 1 = 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0 \oplus 0 = 0$$

$\otimes$ denotes term by term multiplication,

e.g. $\underset{\sim}{R} \otimes \underset{\sim}{S} = \{r_k \, s_k\}$

$\leftrightarrow$ or $\overset{FT}{\leftrightarrow}$ denotes the Fourier transform operation:

$$s(t) \overset{FT}{\leftrightarrow} S(\omega) \quad \text{where} \quad S(\omega) = \mathcal{F}[s(t)]$$

$\overset{WT}{\leftrightarrow}$ denotes the (discrete) Walsh transform operation:

$$\{f_j\} \overset{WT}{\leftrightarrow} \{F_k\} \quad \text{where} \quad \{F_k\} = \mathcal{W}[\{f_j\}]$$

$\overset{zT}{\leftrightarrow}$ denotes the z transform operation:

$$\{r_k\} \overset{zT}{\leftrightarrow} R(z)$$

PREFACE

The work described in this thesis had its genesis in an afternoon tea conversation between Prof. R.H.T. Bates and W.K. Kennedy, of the Electrical Engineering Department in the University of Canterbury, Christchurch, New Zealand, in late 1971. This discussion led to an undergraduate project which involved seven students, including the author, during 1972. An ancient piano, purchased for $50, was fitted with switches, filled with electronics, and connected to the Department's EAI 590 Hybrid computer. Software was written to periodically sample the keyboard switches and to display the resulting table of note parameters in a primitive form of traditional music notation. Unfortunately, the original system never worked because of noise problems associated with the analogue keyboard switch encoding system used (digital input facilities were not available). Despite this drawback the primary objective of establishing the feasibility of connecting a keyboard instrument to a computer for music input was realised.

During November and December of the 1972 vacation I continued work on the input and display system. A digital input device controller was designed, built and connected to the existing RTL I/O bus interface. The piano became a peripheral device of the EAI 640 digital computer. Major software changes were also made, and on Christmas Eve 1972 several appropriate tunes were played and subsequently displayed in our crude conventional notation.

About this time M.R. Lamb (a musician with First Class Honours in Mathematics) became interested in our system. Early in 1973 he enrolled as a Ph.D. student in the Electrical Engineering Department under Prof. R.H.T. Bates, to work on the application of computers to music theory, music teaching and psychoacoustics. A thesis arising from his research is now in preparation.

During 1973 the development of the music input/output and display system continued. The old piano was retired and a small electronic organ purchased. A three word, 48 bit I/O digital interface was developed by the author with help from W.K. Kennedy, and was constructed with the assistance of M.R. Surety, a Departmental Technician. This permitted both the recording and subsequent playback of music over a four octave range. The display notation MOD was also developed during this period.

In late 1973 I transferred from the M.E. (Master of Engineering) course to commence the Ph.D., with Prof. R.H.T. Bates and W.K. Kennedy as joint supervisors. This allowed my research interests to broaden, and I started to work on pitch estimation of acoustic speech and music signals.

During 1973 and 1974 we had weekly discussions with J.E. Cousins of the Music Department in the University of Canterbury, and later extended our contact there to include D.F. Sell. While these meetings were aimed mainly at Lamb's research, they provided me with a useful theoretical background to augment six years as an amateur clarinettist. They also provided direct stimulation for the digital

synthesis system described in Part 2 of this thesis.

About mid 1974 Susan D. Frykberg began studying the musical possibilities of our system, and soon started to implement her own thoughts on how a computer can be used both as a composition aid and as an idea generator. Her work, like much of Lamb's, uses the organ system as a music interface between the creative human and the computer. I have made much use of their comments and suggestions in the development and refinement of the system.

The next members of our team, R.J. Howarth and R.G. Vaughan, entered the scene in 1975 when they worked on digital synthesis of organ voices as an undergraduate project. As with the original piano project, they extensively revised and reconstructed most of what they had built for their project during two months' vacation work. Vaughan continued the development of both hardware and software as an M.E. project during 1976. Simultaneously, Howarth developed for his M.E. a software package which permits elegant music copy to be produced by the CALCOMP plotter in the University of Canterbury Computer Centre. This music display package operates in "batch processing" mode, and uses as input a paper tape which is punched by the interactive conventional music notation transcription module of the organ system when editing has been completed. Vaughan's work is now (1977) being extended by P.J. Cheyne, also as an M.E. project.

The main contributions of this thesis are in music transcription, display and editing using conventional notation, and in the extension of pitch estimation from the

well researched area of speech analysis into acoustic music analysis.

Much has been done elsewhere on conventional music display and printing by computer. Nevertheless, previous efforts to develop music transcription systems which accept played keyboard or acoustic music as input and produce conventional written music as output have been relatively unsuccessful. While our work is not complete, it is a significant advance and has potentially important practical applications in the music publishing industry. In automatic pitch analysis the contribution presented in this thesis is twofold. First, I review comprehensively those pitch estimation techniques in speech processing which have appeared since McKinney's (1965) survey - the emphasis here is on digital methods, and the inherent and practical difficulties encountered when many of these techniques are applied to wideband music signals are identified. Second, I present a new algorithm which works for a wider class of signals than previously reported methods, and which is computationally efficient. Unlike the most useful of the existing heuristic methods (which are also efficient from a computational viewpoint) the new algorithm is analytically relatable to autocorrelation analysis. This algorithm was developed jointly with Prof. R.H.T. Bates.

The scope of this thesis is introduced in detail in Chapter 1, in which it is made clear which parts are review material and which parts describe original contributions.

The following papers, relevant to the subject matter of this thesis, have been prepared for publication:-

Tucker W.H., Lamb M.R., Howarth R.J., Vaughan R.G., Kennedy W.K., Frykberg S.D., and Bates R.H.T. (1975). Computerised musicianship aids. Presented at National Electronics Conference (NELCON '75), Wellington, New Zealand, August 1975.

Tucker W.H. and Bates R.H.T. (1977). Efficient pitch estimation for speech and music. Electron. Lett., vol. 13(12), pp.357-358.

Tucker W.H., Bates R.H.T., Frykberg S.D., Howarth R.J., Kennedy W.K., Lamb M.R., and Vaughan R.G. (1977). An interactive aid for musicians. Accepted by: Int. Journ. Man-Machine Studies.

Tucker W.H. and Bates R.H.T. (1977). Pitch estimation algorithms for speech and music. Submitted to: IEEE Trans. Acoust., Speech, Signal Processing.

Bates R.H.T., Cousins J.E., Frykberg S.D., Kennedy W.K., Lamb M.R., Sell D.F., and Tucker W.H. (1975). The computer as an aid in music research and education. Submitted to: Yearbook of the National Consortium for Computer Based Music Instruction.

CHAPTER 1

INTRODUCTION

The application of digital computing techniques to the fine arts, and especially to music, has received increasing attention over the last decade. Music in particular is well suited to storage in and manipulation by computer, because of its well-defined components, rules, and formal structure. Application areas which have been reported include computer generated music composition (Hiller and Isaacson, 1959; Xenakis, 1971), computer controlled music performance and sound synthesis (Zinovieff, 1968; Mathews, Moore and Risset, 1974), music printing by computer (Hiller and Baker, 1965; Böker-Heil, 1972; Smith, 1973), and computer aided teaching of both music theory and practice (Hofstetter, 1976). In addition, the computer has been used to carry out analyses spanning a broad spectrum of music theory and musicology (Heckman, 1967; Brook, 1970; Lincoln, 1970). The potential usefulness of man-machine interaction in creative musical tasks is now widely recognised.

A major obstacle which seems to inhibit the usefulness of many existing systems lies in the design of the man-machine interface. For example, a frequently used approach to specifying music for entry into a computer is to manually encode the music score into a string of

alphanumeric characters. Numerous encoding "languages" have been developed for this purpose (Byrd, 1974; Styles, 1974). However, such an encoding language is both unnatural and inconvenient for most musicians and composers. As Smoliar (1973) has observed, a primary desideratum of any computer-music facility is that the user (musician, composer, teacher or student) be able to approach it with a sense of convenience and flexibility. In addition, since creativity is the province of the musician and not of the computer, it is now recognised that interactive "hands on" computer music systems are more effective than systems which operate in "batch processing" mode (cf. Smoliar, 1973). Thus the prerequisites for a successful system must include convenience and flexibility for the user, the incorporation of conventional musical instruments, the use of standard (or easily learned) music notations, and rapid execution of a wide range of musical tasks (cf. Pulfer, 1971; Fredlund and Sampson, 1973).

This thesis considers the problems associated with the encoding, storage, manipulation and presentation of music in an interactive computing system. It discusses the development of data structures, interface hardware and software, and the organisation of a system which permits musicians to interact with both the acoustic and symbolic manifestations of aural and written music. The applications of this system to music teaching, music composition, real-time performance, and music typesetting are mentioned. These applications are discussed in more detail in a companion thesis by the author's colleague Lamb (1977).

Lamb also considers the use of this system as a "musician-computer interface" in the broader context of music-theoretical and musicological processing by computer - topics which are not mentioned in this thesis. Thus, the work described herein is oriented essentially towards the electrical engineering disciplines of interactive computing software systems, interface hardware, electronic sound synthesis, and acoustic signal processing.

This thesis is organised in three main parts. Part 1 (Chapters 2 to 4) considers the application of computers to music-oriented tasks, and describes the system developed at the University of Canterbury. In Part 2 (Chapters 5 and 6) the use of electronic techniques for sound synthesis is discussed, with particular emphasis on digital rather than analogue methods. The theme of Part 3 (Chapters 7 to 9) is the measurement of pitch trajectories from acoustic speech and music signals. New material is introduced in Chapters 3, 4, 6 and 9.

Chapter 2 briefly reviews existing computer-music facilities. The emphasis here is on interactive or batch processing systems rather than on techniques. This review serves to place in perspective the work described in the remainder of the thesis.

Chapter 3 describes the organisation and functioning of the component modules of our interactive aid for musicians. This system permits the "recording" of music played on an electronic organ, and its subsequent playback. Facilities for the display and editing of this music are provided using a graphical display unit. These facilities

use a positional notation of a kind often used by contemporary composers. Additional editing facilities which permit editing to be carried out in the aural rather than visual music domain are also provided. The material presented here is original, and covers the design and development of the overall system and its hardware and software components.

Chapter 4 describes an extension of this system, which permits music recorded from the organ keyboard to be transcribed and displayed in conventional music notation. An essential feature of this process is the inclusion of flexible editing facilities which permit the display presentation as well as content to be altered manually. Examples are presented which illustrate the various stages required for the preparation of music typescript, and which indicate the kinds of results which can be achieved. New material is introduced throughout this chapter.

Chapter 5 reviews the field of electronic sound synthesis. Both analogue and digital techniques are discussed, but the emphasis is on the latter. It is advocated that a digital computer be used to control dedicated synthesiser hardware rather than to perform the actual sound synthesis. Linear prediction (which is commonly used in speech synthesis) is suggested as a potentially useful synthesis technique (Section 5.4).

A computer-controlled digital synthesis system is described in Chapter 6. This system permits the generation of a wide range of timbres, and complements and extends the organ playback facility described in Chapter 3. New

material is presented throughout. Sections 6.1 and 6.2 describe the development of the overall system, while Sections 6.3 to 6.5 cover hardware development. Software aspects are discussed in Sections 6.6 and 6.7.

Chapter 7 presents a detailed review of pitch trajectory estimation techniques. The perception of pitch and the measurement of its physical correlates are outlined. The essential differences between speech and musical signals are discussed. A variety of pitch estimation techniques which have been applied to speech are described, and are assessed with respect to their suitability for wide-band musical signals. The material presented here is mainly review, although the analysis of the interdependence between signal sampling rate, pitch frequency resolution, and pitch frequency range given in Section 7.4.2 is original.

In Chapter 8 is reviewed a number of recent transform techniques which permit the efficient numerical evaluation of signal correlations. This review includes the number-theoretic transforms and methods based on the fast Walsh transform, as well as recent improvements to the well-known fast Fourier transform algorithm. It is concluded that these methods do not offer sufficient computational superiority over the conventional fast Fourier transform to alter the conclusion of Chapter 7 - namely that the use of pitch estimation methods which use correlation techniques are not practicable for signals whose pitch trajectories span more than two or three octaves.

Chapter 9 describes a new pitch estimation algorithm which operates in the time domain by recognising recurring

"features" of the signal waveform. This algorithm is compared with the well-known Gold and Rabiner algorithm, and comparative results are presented. These results show that the new method works for signals for which the Gold and Rabiner method is unsuccessful. The new method is shown to be analytically relatable to autocorrelation analysis. New material is presented throughout. Factors influencing the development of the algorithm are discussed in Section 9.1. A general mathematical foundation is given in Section 9.2. The implementation of the Gold and Rabiner algorithm is described in Section 9.3. More mathematical material is presented in Sections 9.4 and 9.5, where the new algorithm is described and is related quantitatively to autocorrelation analysis. Results are given in Section 9.6. An algorithm which quantises a continuous pitch trajectory into the discrete form required by the music transcription system (cf. Chapters 3 and 4) is described in Section 9.7.

Chapter 10 summarises the main conclusions of this thesis and suggests areas of further research.

PART 1

INTERACTIVE AIDS FOR MUSICIANS

CHAPTER 2

COMPUTER AIDS FOR MUSICIANS - A REVIEW

The application of digital computing techniques to music can be classified into four areas:

(i) The computer as a composition aid.

(ii) Computer-controlled music performance and sound synthesis.

(iii) Music printing by computer.

(iv) Computer-aided teaching.

This chapter reviews the literature pertaining to these application areas. Other application areas (such as the use of linguistics and artificial intelligence for the analysis of musical style, structure and content, or the generation of thematic catalogues) which belong to the wider realm of music theory and musicology are not considered here. These latter topics are discussed elsewhere (Heckman, 1967; Brook, 1970; Lincoln, 1970; Laske, 1973; Smoliar, 1974) and are reviewed by Lamb (1977). It is worth mentioning here the articles by Bowles (1970) and Kassler and Howe (1975) which provide useful introductions to the field of computers and music, music theory and musicology. Comprehensive bibliographies are provided by Brook (1970) and by Kassler and Howe (1975).

2.1 THE COMPUTER AND COMPOSITION

Computers have been used to "compose" music almost since the inception of digital computing. An extensive review up to 1970 is given by Hiller (1970).

Early work on computer composition used a statistical approach. In 1949, J.R. Pierce and M.E. Shannon created a sequence of notes by making random selections from a table of allowable root chords in the key of C (Pierce, 1961). Brooks *et al.* (1957, 1958) used a Monte Carlo method in conjunction with the measured statistics of existing music to generate new hymn tunes. Since the approach used by Brooks *et al.* is typical of that used by many other early workers (cf. Pinkerton, 1956; Hiller, 1959; Olson and Belar, 1961; Hiller and Baker, 1964) it is worth describing more fully. A selection of 37 existing hymn tunes (transposed into the key of C) was entered into the computer. Each tune is assumed to consist of a sequence of notes representative of a Markov process of order m. The statistics of the representative group of tunes is accordingly evaluated. A new tune is generated by constructing a sequence of randomly chosen notes such that the statistical parameters of the new sequence and the representative tunes are identical. Additional heuristics are included to handle rests, ties and durational values, and to constrain the new sequence to end on the note C. The effect of the order m of the Markovian process on the "acceptability" of the new tunes was subjectively evaluated, for m ranging from 1 to 8 (as m increases, the "context" of each note becomes

increasingly important). A value of m intermediate between the extreme values 1 and 8 was judged to yield tunes both hymnic and novel, although the criteria for this decision are not specified.

Hiller and Isaacson (1959) used a similar algorithmic procedure to compose music - the Illiac Suite for string quartet is a well-known example. Lambert Meertens' Quartet (1968) is also worthy of mention as an example of algorithmically composed music in the "classical" style.

Xenakis (1963, 1971) introduced the concept of "stochastic music" in which particular attributes such as pitch and duration are generated randomly, but in accordance with some predetermined statistical distribution. A distinguishing feature of this work is that statistical distributions from fields other than music are often used, and that no attempt is made to model existing musical styles.

The composition techniques outlined so far are all oriented towards "batch processing", in the sense that the computer generates the entire musical passage without intermediate human intervention or assistance. In the late 1960's some attempt was made to overcome the perceived deficiencies of the early "compositions" by incorporating human value judgements interactively into the composition process. Thus the computer is used as a "composer's assistant".

Pulfer (1970) describes an interactive facility designed specifically to aid the composition, arrangement and "live" production of music. This system incorporates an organ keyboard, several graphical display units, a light

pen, and an analogue positioning wheel. The user enters a melody by playing the organ keyboard and using the positioning wheel and teletypewriter to specify note pitches, durations and rests (the procedures used are not clearly described). The resulting music is then displayed on a graphical display unit, using a notation similar to conventional music notation. Control information such as the attack and decay of notes, the key, and logical boundaries of portions of the melody can be specified. The speed and loudness of the entire piece can also be defined, and the loudness of individual notes can be specified. Editing facilities permit a melody to be modified - thus individual notes or control parameters may be deleted, inserted, or overwritten. The waveform which constitutes one period of the sound desired can also be specified by sketching an amplitude versus time curve on the display screen using the light pen.

The melody and waveforms specified in this manner can be "played" using real time software digital sound synthesis. An "arrangement" facility permits the user to transpose and concatenate individual sections of a melody, as well as to specify the envelopes and waveshapes required for each note or section. Melodies, waveshapes and arrangements may be stored for subsequent recall.

An interesting feature of Pulfer's work is that considerable emphasis is placed on the design and evaluation of the man-machine interface - thus "the intent is not to develop a good music facility, but to find out more about the interactive problems in a creative environment".

This presumably explains several significant limitations in the system. Firstly, only a melody or single musical line can be handled. This limitation places a serious restriction on the usefulness of the system, and arises as a consequence of the software sound synthesis technique used (cf. Section 5.3). Secondly, and of lesser importance, pitch values are constrained to those of the chromatic scale, and time intervals (such as note durations) are constrained to multiples of a 1/32 note.

Smoliar (1973) describes a composition aid which is similar in concept to Pulfer's system. However, while Smoliar acknowledges the desirability of using a piano-like keyboard as an interactive device, his implementation falls short of this ideal and requires that a teletypewriter be used instead. Smoliar also uses software sound synthesis, with the result that only a limited range of sounds can be produced in real time. It is worth pointing out that Smoliar is concerned primarily with interaction in the aural music domain, whereas Pulfer considers interaction in both the aural music and written music domains. These topics are discussed in more detail in Sections 2.2 and 2.3 respectively.

An entirely different approach to computer-aided music composition is that which uses "artificial intelligence" to attempt to determine and to explicate the various decision-making processes used by human composers. Reitman and Sanchez (Reitman, 1960) incorporate the "General Problem Solver" (Newall, Shaw and Simon, 1958, 1960) into a set of heuristics which was developed from extended

observation of a composer at work. Laske (1974) uses a similar approach but automates the observation process by using a computer to monitor the actions of a composer working with a special interactive composition aid. Winograd (1968), Smoliar (1972) and Truax (1973a, 1974) apply the concepts of computational linguistics to composition.

Another approach to music composition is analysis-synthesis of musical style, in which the music rather than the composition process is considered (Jackson, 1967; Lamb, 1977). Work in this area relies on both music heuristics and on well-determined musical rules such as those codified by Lovelock (1946) and Piston (1947). The analysis techniques developed by Roller (1965), Forte (1966), Brender and Brender (1967), Suchoff (1968), Mendel (1969), Fuller (1970), Lefkoff (1970), Youngblood (1970) and Patrick (1974) are also relevant here.

A more detailed discussion of the application of computers to composition is given by Frykberg and Bates (1977).

2.2 COMPUTER CONTROLLED MUSIC PERFORMANCE AND SOUND SYNTHESIS

Computer-controlled music performance can be achieved in any of three ways:

(i) By direct generation of the acoustic signal, for example using digital to analogue converters.

(ii) Using electronic synthesisers or other electro-acoustic "instruments".

(iii) Using conventional acoustic instruments.

It is convenient to divide the information, that is (in general) required for a performance, into two distinct classes - namely "instrument performance" and "instrument design" (Smoliar, 1973a, b). The "instrument performance" information specifies those parameters contained in a conventional music score - viz. pitch, note onset time, duration, and loudness. This information is required for a computer-controlled performance using any of the three approaches listed above. "Instrument design" information specifies the timbre of the sounds, and is required for the first two categories. The parameters which are required depend intimately upon the organisation of the synthesis system. The use of electronic techniques for sound synthesis is considered in detail in Part 2 of this thesis, and is not discussed further here.

2.3 MUSIC PRINTING BY COMPUTER

The development of systems which produce written music in conventional music notation is closely linked historically to the development of methods for specifying the parameters of written music in machine-readable form. Both these topics are discussed in this Section, because of their close interdependence.

Kassler (1977) presents an eloquent argument for producing written music by computer. Kassler shows that master pages (suitable for subsequent reproduction, for example by photo lithography) can be produced by computer for less cost than if the traditional labour-intensive

methods are used. These latter include engraving, music typing, and autography (i.e. reproduction from hand-copied music) - see Howarth (1977a) for a review of traditional music printing methods. In addition to this economic advantage, Kassler points out a number of useful by-products which may be automatically produced by a computer-based music printing system at little additional cost. These include:

(i) Music output in Braille.

(ii) Music output in a variety of keys and pitch ranges for different instruments, including transposing and non-transposing instruments. Similarly the various vocal ranges (e.g., soprano or bass) can be accommodated.

(iii) Music output in a variety of editions (e.g., with or without fingering indications).

(iv) "Reductions" of full scores (e.g., of orchestral works) to piano scores.

(v) Electronic sound synthesis of the music to permit "proofhearing" as a supplement to proofreading.

(vi) Catalogues of themes or incipits (an incipit is the initial segment of a piece of music).

(vii) Computation of chordal indicators (e.g., "G7") which often accompany printed songs.

(viii) Anthologies consisting of music which obeys certain well-defined selection criteria.

Various methods of specifying written music parameters and subsequently producing a score are now discussed.

2.3.1 Written Music Input

Comparatively little work seems to have been done on direct music input from printed scores using optical scanners. Presumably this lack of attention is due to the fact that such a system is likely to be both expensive and limited in its usefulness, since it is applicable only to music for which a well-printed score already exists (Styles, 1974). Another contributing factor could be the lack of potential market (Kassler, 1977). The only significant work in this area seems to be that of Prerau (1970, 1971), who describes a system (called DO-RE-MI, for Digital Optical-Recognizer of Engraved-Music Input) which performs optical recognition of a range of music symbols. The output of DO-RE-MI is an alphanumeric coding in the Ford-Columbia language (which is described below). Prerau's system was tested on about 20 bars of a Breitkoph & Härtel publication of Mozart's Twelve Duets for Two Wind Instruments, K. 487 whose symbols it recognised correctly. While this passage does not include all the symbols encountered in conventional music notation, it contains a sufficiently large subset to suggest the technical feasibility of widespread optical music input. It is worth mentioning that Prerau cites earlier work by Pruslin (1967), although this latter is comparatively limited in the range of symbols it can recognise.

Numerous "languages" have been devised to represent conventional music notation using alphanumeric character strings. An extensive description of many of these languages is given by Brook (1970), while Byrd (1974) and

Styles (1974) also provide useful reviews. The basic problem facing all designers of input languages is that "one is trying to express a two-dimensional assembly of items from a very rich character set by means of a uni-dimensional string of symbols from a rather meagre set" (Styles, 1974). It should also be pointed out that conventional music notation contains much information that is redundant and which can be omitted from the corresponding alphanumeric code (Böker-Heil, 1972), although the elimination of such redundancies will of course require an increase in the complexity of the processing programs (Styles, 1974). This point is considered in detail in Chapter 4 (cf. Sections 4.1, 4.3 and 4.5).

One of the best known music encoding languages is DARMS (also called the Ford-Columbia language). DARMS was developed in the mid 1960's as part of a computer-controlled photocomposition system for music printing, under the direction of Stefan Bauer-Mengelberg and Melvin Ferentz (Bauer-Mengelberg, 1970). DARMS is designed to permit the encoding of complete scores, in full detail, and appears to be the only language which attempts to represent all of the characteristics of printed music (Styles, 1974). It is unusual in that it defines pitch by numbering the stave lines (from 00 to 49) rather than by using the letters A to G with suitable octave indicators - treble C (i.e. C5) is thus 26, and values in the twenties may omit the leading 2. An excursion of the same instrument on to a different staff invokes the addition of a multiple of 50 to the pitch value. Durations are denoted by the letters B (Breve), W (Whole

note), H, etc. Stem directions can be assigned by U, D, or % (i.e., one of each). Slurs and ties can be tagged by numbers, so that their ends can be associated with the correct notes.

A Plaine and Easie Code System for Musicke (Brook and Gould, 1964; Brook, 1970) was designed for the storage of short sections of music. Pitch is represented by the letters A to G using normal music conventions, and any note defined in this way is assumed to be within a fourth of its predecessor. Pitch excursions larger than a fourth are indicated by apostrophes and commas. Note durations are defined as fractions of a semibreve using a numerical representation (e.g., 1, 2, 4, 8 denote semibreve, minim, crotchet, quaver, respectively). Triplets and other unusual rhythmic combinations are defined by enclosing in parentheses the durations as written (e.g., quavers), and preceding the list with the total duration of the combination of notes. Such groups may be nested. Repeated rhythmic patterns or entire bars are conveniently handled without requiring the initial character string to be duplicated.

IML (Robinson, 1967) encodes one staff at a time. The letters A to G are used to define pitch. Notes are assumed to lie between the bottom and fourth stave lines, unless octave shifts (denoted by U or D) indicate otherwise. An octave shift, once indicated, remains operative until countermanded or until the end of the current bar is reached. Pitches are defined according to the conventions used for the treble clef, even if another clef is actually used.

Durations are denoted by digits which represent the corresponding fraction of a semibreve. Tied notes are enclosed between a pair of asterisks - the possibility of simultaneous ties is not mentioned. The IML assembler converts pitches to those of the clef actually used, and performs simple grammatical checks on the coded character string. An additional programming language called MIR (Erickson, 1968; Kassler, 1970) is also available to permit musicians to write programs which interrogate the structure and content of the encoded music data.

Wenker (1970) describes a system (not implemented) which is intended to cater for the special symbols used by ethnomusicologists as well as those used in conventional music notation. The notes between middle C (i.e., C4) and treble B (i.e., B4) are denoted by the corresponding letters, and plus and minus signs are used to indicate octave shifts for notes outside this pitch range. Triplets and similar irregular rhythmic combinations are indicated by parenthesizing the note group, preceding the parenthesized group by an integer which specifies the number of notes in the group, and using the character "3" as the last character in parenthesis. The total real-time duration of the music passage is stored in a separate table. Multiple voices on a single staff are handled - the symbol "Y" between adjacent note descriptions indicates that both notes start simultaneously.

LMT (Regener, 1967) is unusual in that pitch, duration etc. are encoded in separate lists. This requires that control information be duplicated, and suffers from the

possibility of discontinuity of list correlation should small errors be encountered. Nevertheless, a number of advantages are claimed for this system. These include quicker transcription, the potential for using the input character set more effectively (since a given character may be used for several parameters), and greater ease of representing repeated groups of parameters. Notes beamed together are parenthesized, and a number of dots is inserted between those pairs of notes for which some beams are missing (the number of dots corresponds to the number of beams missing).

Ashton (1970) describes a transcription notation which was developed as part of a system for the acquisition, performance and display of music (see also Knowlton, 1971, 1972). Pitches are denoted by the corresponding letter and octave number (e.g., C4 denotes middle C), and durations are specified by the letter corresponding to the fraction of a semibreve. A group of notes enclosed in parentheses is interpreted as a chord, and some scan control is provided to enable shifts to the beginning of a particular bar n (:n:) or to the beginning of the current bar (;).

ALMA (Gould and Logemann, 1970) is derived from the Plaine and Easie Code mentioned above, and is intended for applications in information retrieval. Although ALMA is very sophisticated and therefore difficult to learn, most musical features can be encoded in several ways. Consequently, a subset of the full language is sufficient for many musical scores. For example, a chord may be denoted in three ways - the constituent notes may be

enclosed within a pair of $ characters, the bottom note and the intervals may be defined, or < may be used to shift the point of scan back to the beginning of the preceding note. Thus the triad of C major may be denoted as CEG, C33, or C < E < G. ALMA explicity distinguishes between notes (including chords and rests) and attributes (such as phrasing, accidentals and scan control). The point of scan can be shifted in any direction, relatively and absolutely, in terms of notes or entire bars. Macro-like facilities are available for applying rhythmic sequences and attributes to strings of notes, and for defining recurrent pitch sequences.

Styles (1974) describes an encoding language and defines its canonical syntax in Backus Naur Form. Pitches are defined numerically (1 corresponds to the lowest stave line and 9 to the highest stave line). Notes below or above the staff are indicated by a dot which precedes or follows the pitch number. Durations are specified by the letters L, B, S, M etc. Repetitions of rhythmic or harmonic sequences may be easily notated by enclosing in parentheses a sequence of pitches or durations. Groups defined in this way can be tagged (for subsequent recall), nested, or repeated (by following the closing parenthesis by an integer which specifies the number of repetitions). Beaming is indicated by enclosing the operator = between those notes so connected, and using commas to denote missing beams. Notes or groups which are enclosed within square brackets are interpreted as starting simultaneously, so that chords and phrases which begin together can be easily defined. Facilities are provided for full scan control, so that the

scan pointer may be shifted in any direction to an absolute position, or by a specified amount relative to the current position.

The coding scheme of Lefkoff (1967a) which uses a special coding sheet and that of Duerrenmatt, Gould and La Rue (1970) which uses mark-sense cards should also be mentioned here, although these are essentially two-dimensional positional codes rather than alphanumeric codes.

In an attempt to overcome many of the difficulties (already mentioned) which are inherent in describing two-dimensional written music by a one-dimensional character string, Fredlund and Sampson (1973) developed a direct, two-dimensional input system using a graphical display unit. This input system displays a "menu" or list of commands. The required command is indicated by pointing to it using a light pen. A staff is generated at a specified vertical position by indicating the command STAFF, and then pointing to one of a number of vertically separated asterisks. Music symbols such as notes, rests, or accidentals can be placed on the staff by pointing to the required prototype symbol (which is displayed as part of the "menu") and then specifying the position at which that symbol is required. Scores generated in this manner can be stored for subsequent recall, copied on to different staffs, transposed, copied with a pitch inversion, or copied with a retrograde inversion. A disadvantage of the system is that only a limited number of symbols have been implemented. Also, the positions at which each symbol may be placed are relatively widely separated, so that musical punctuation and

justification (cf. Section 4.5) are not considered. Styles (1974) comments that this approach to music input is also prone to divorcing the data from its musical significance.

Smith (1973) describes an interactive music input system called MSS which uses a multi-stage entry process. In the first stage the clef, time and key signatures and note pitches are entered on an alphanumeric keyboard. Note pitches are specified by their letter names with an octave number, together with the accidental (F, S or N) if required. A colon following a note indicates that the note will appear in the same rhythmic position as the previous note, so as to produce a chord. In the second stage the horizontal positions corresponding to the start and end of the current note sequence are specified. The notes are then displayed as equally-spaced crotchets on a graphical display unit. Next, the duration of each note is specified (4, 8 correspond to quarter and eighth notes, etc.). The displayed notes are correspondingly altered. Beams are added next, then accents and staccato dots, and finally slurs and ties. The display is updated at the end of each input stage. Comprehensive editing facilities are provided to permit alteration to individual notes or chords or to duplicate groups of notes on to different staff positions. The editing facilities also permit musical punctuation and line justification to be achieved. An unusual feature is a facility which permits the user to draw special symbols using a light pen. The shapes of such special symbols may be subsequently altered by editing. When a special symbol is used in a score it may be inverted, reversed or scaled in size.

Another approach to music entry for computer processing is the development of special-purpose input terminals. Hiller and Baker (1965) used a specially-modified electromechanical music typewriter to produce a punched paper tape as the music is "typed" manually. A special feature of the typewriter is the provision of horizontal and vertical platen shifts in both the forward and reverse directions. The tape produced by this machine is read by the computer for subsequent processing (e.g., error correction, format layout and line justification). A new tape is then punched by the computer, and fed back into the typewriter to produce the final music copy.

Dal Molin (1973, 1977) describes a system which has been used in conjunction with a special photo-typesetter to produce printed music commercially since 1970. It is worth commenting that Kassler (1977) has recommended that Dal Molin's system be used in a proposed commercial computer-assisted music printing centre in Australia.

Dal Molin's terminal consists of three separate sets of buttons, each set being used for a particular class of information. The desired staff-position is specified by pushing one of 28 buttons, which are arranged in four "octaves" of 7 staff-positions each. A keyboard which is essentially that of a music typewriter is used to enter the required symbol at the specified staff-position, or to enter alphanumeric text. A third set of 12 buttons (called the "command pad") is used to specify time and key signatures as well as control information.

The first-generation terminal - designated the PCS-300 - produced a punched paper tape using a specially constructed electromechanical music typewriter. This is now being replaced by an all-electronic terminal (the PCS-500) which incorporates a visual display unit to show the operator the music as it is being entered.

The concept of automatic transcription of written music from its aural performance is not new, and its implementation has been considered by many musicians (Ashton, 1970; Knowlton, 1971). The numerous problems which are encountered in the transcription process are identified and discussed in detail in Chapter 4 (cf. Sections 4.1, 4.2, 4.4 and 4.5).

The work of Ashton (1970) and Knowlton (1971, 1972) is probably the best known attempt to implement automatic music transcription from a keyboard instrument. Their system "captures" the music as it is played on an electronic organ, by sampling a set of switches which are associated with the keyboard (cf. Section 3.1). A constant sampling rate of 20 keyboard scans per second is used to construct a table of note start times, pitches and durations (cf. Section 3.4). Note durations are converted from their actual played durations (in seconds) to the corresponding values (in fractions of a semibreve) by assuming that the real-time duration of a beat is constant. A coarse time-quantisation is used to truncate the temporal variations which are inevitably encountered. The limitations of this approach are pointed out in Chapter 4 (cf. Sections 4.1 and 4.4).

Longuet-Higgins (1976) considers music transcription from the viewpoint of the cognitive psychologist, and points out the similarity between many of the problems of melody perception and speech perception. Rhythm is treated by considering as nodes of a "tree" structure the metrical units which constitute the rhythm. The onset of each note is used to predict the time of onset of the following note at each level of the tree structure. An interesting feature of this work is that synchopation, triplets etc. are explicitly catered for, although obviously a precise performance is required if such rhythmic features are to be recognised correctly.

Mars and Cattanach (1977) describe a transcription system which is remarkably similar in both its design and performance to the system developed as a Final Year Undergraduate project in 1972 by the author and others, and subsequently abandoned (see Section 4.1). Kassler (1977) also mentions a transcription system which is being developed at the Australian National University, although nothing seems to have been published concerning this project yet.

The ambitious work of Moorer (1975) should be mentioned here. Moorer considers the problem of transcribing into conventional music notation the acoustic signal of polyphonic aural music (i.e. aural music which consists of chords). His main concern is with the estimation of pitch, and comparatively little attention seems to have been paid to the "roundoff" problem (Section 4.4) which is encountered in the estimation of

durational values. Smith's (1973) MSS system is used to produce the final music copy.

2.3.2 Written Music Display

The output devices that have been used to produce master pages of music are of two kinds: those in which the individual musical symbols such as accidentals, note heads, etc. are built into the hardware; and those in which the symbol shapes are encoded in the software which controls the hardware. The first category includes the music typewriter, phototypesetter, and special character-set impact printer. The second category includes the cathode ray tube, mechanical (i.e. moving pen) plotters, electrostatic plotters and xerographic printers. Byrd (1974) points out the advantages of using devices in the latter group, since some musical symbols (e.g. ties, slurs and beams) are of arbitrary size and may be inclined at arbitrary angles to the stave lines. The relative advantages of cathode ray tubes, mechanical plotts and phototypesetters are considered further in Sections 4.7 and 4.9. It is sufficient here to state that both phototypesetters and mechanical plotters have been used to produce music whose visual quality is of similar standard to that of engraved music (Kassler, 1977). It is also worth pointing out that while the visual quality depends primarily on the hardware and/or software used to produce each symbol, other aspects such as the horizontal spacing of notes (i.e., musical punctuation) are important too. The former can be improved by using better output devices whose resolution is finer (or equivalently by drawing the music at a larger size

and using photographic reduction). However, musical punctuation is determined by the controlling program and is a complex logical problem (see Chapter 4). Consequently, improvement at this level is comparatively difficult to implement.

Early work in computer-assisted music printing includes that of Hiller and Baker (1965) who use an electro-mechanical music typewriter controlled by paper tape, and Lincoln (1970) who uses a line printer with a special character set. Gabura (1967), Raskin (1967), Böker-Heil (1972), Byrd (1974), Fellgett (1974) and Styles (1974) have all developed batch processing systems which use a mechanical plotter as the output device. Interactive systems which have been reported include Cantor's (1971) music editor (which uses a cathode ray tube display) and MSS (Smith, 1973). The operation of MSS is outlined in Section 2.3.1. The interactive creation and editing of the score is performed using a cathode ray tube display, while the final hard copy is produced at about 1.5 times the required final size using a Calcomp plotter, and is subsequently reduced photographically. Smith also uses a Xerox Graphics Printer which is much faster than the plotter, but achieves better visual quality with the latter. The Dataland system mentioned by Kassler (1977) also uses a mechanical plotter.

The Music Reprographics system (Dal Molin, 1973, 1977; Kassler, 1977) has been producing music copy commercially since 1970. The special purpose input terminals used are described in Section 2.3.1. The music

copy is produced using a special phototypesetter (see Section 4.9).

2.4 COMPUTER AIDED TEACHING

Hofstetter (1976) provides an extensive review of the application of computers to music teaching. Lamb (1977) also considers this topic in detail. The treatment here is cursory, because teaching is of only peripheral relevance in this thesis.

Hofstetter divides computer-aided music teaching into two main categories, namely those in which the computer is used to present instructional material, and those in which the computer serves as an interactive tool. Instructional programs are helping students to learn instrumental techniques (Diehl, 1971, 1973; Peters, 1974, 1975), the fundamentals of music theory (Placek, 1974), ear-training (Kuhn, 1974; Hofstetter, 1975; Lamb, 1977) and set theory (Forte, 1973). Interactive computer-aided instruction systems which have been reported include those of Baker (1971) and Arenson (1975), while Finch (1972) reviews early work in this area.

CHAPTER 3

AN INTERACTIVE "PIANO TYPEWRITER" SYSTEM

3.1 PREAMBLE

A flexible, interactive music input, output and display system with comprehensive editing and manipulative facilities has been developed, using the EAI 590 hybrid computer facilities in the Electrical Engineering Department of the University of Canterbury.

The system is a complete entity in its own right, and is used for teaching, composing, and producing music typescript. In addition, it is a useful interface for a number of other computer music applications. Lamb (1977) uses it in his interactive teaching and musicological analysis systems. It forms an integral part of a composition aid system being developed by Susan D. Frykberg. Vaughan (1977) in collaboration with Miss Frykberg and the author, has added to it the digital synthesiser discussed in Chapter 6. Howarth (1977a) has developed music display software for the B6718 computer and Calcomp plotter in the Computer Centre of the University of Canterbury. This complements the author's display system, and is intended as a high quality off-line plotting system to overcome hardware limitations encountered with the EAI 590 graphics facility.

The "Piano Typewriter" development began as an undergraduate project in 1972, with the aim of interfacing

a piano to the EAI 590 hybrid computer to provide music transcription facilities. It was felt that music parameters such as note start time, pitch and duration could be conveniently entered into the computer memory by periodically sampling a set of switches fitted to the piano keyboard. These note parameters could then be used to generate a music display. Subsequent editing procedures would permit changes to both the display content (e.g. note pitch and duration values) and form (e.g. stem directions and lengths, note spatial positions). Seven students were involved, under the joint supervision of Prof. R.H.T. Bates and W.K. Kennedy. Reports by Baird (1972), Balfour (1972), Hosking (1972), Mukwamataba (1972), Roche (1972), Tucker (1972) and Wells (1972) describe the work completed during this undergraduate project. Although the resulting system never worked as a complete unit, primarily because of noise problems encountered with the analogue keyboard encoding and input system used, it did convince us of the feasibility of our approach to music input for computer processing. It also strengthened our initial convictions that the system design should be oriented towards interactive rather than batch processing use.

Subsequent hardware and software development by the author has achieved reliable music input and playback facilities using a small electronic organ. It was found to be convenient to be able to display and edit recorded music information in either conventional notation or in a modern notation of a kind often used by contemporary composers. These two notations are henceforth denoted by TRAD and MOD

respectively, and are described in Chapter 4 and Section 3.7 respectively. It is worth noting here that considerable effort has been required to develop the TRAD transcription, display and editing system to the stage where it can produce music typescript of an aesthetic and professional standard acceptable to musicians.

In this chapter the keyboard music recording, playback and MOD display and editing portions of the system are discussed. Both hardware and software aspects are considered, together with the organisation and multi-phase implementation of a large interactive system on a small digital computer. The TRAD notation transcription and editing system is discussed in Chapter 4 and is illustrated there with examples.

3.2 SYSTEM OVERVIEW

The EAI 590 Hybrid computing system of the Electrical Engineering Department consists of an EAI 640 digital computer, a small EAI 580 analogue computer, and a hybrid interface between the two units. The digital computer has 16K , 16 bit words of magnetic core memory, a fixed head disc (360K words memory) and supplementary magnetic and paper tape storage facilities. An electronic organ whose keyboard is fitted with switches - one for each note - is connected to the EAI 640 by a digital interface which permits the status of each organ key to be interrogated. Interactive graphics is provided by a Tektronix 611 storage oscilloscope, a joystick which controls a light spot on the oscilloscope screen, and a hard-copy unit (Tektronix 4601).

An advantage of a storage oscilloscope over a refresh display is that finely detailed displays may be generated without flicker (see for example Ritchie and Turner, 1975). A DEC-writer teletype permits on-line interaction between the user and system, through interpretive controlling software. This is summarised in Figure 3.1.

The organ digital interface is provided with a latch memory, and the organ is arranged so that a note sounds if either a key is depressed or the corresponding latch bit is set. The status of each key is sampled sequentially at a 100 Hz rate. This allows a table of note pitches, start times and durations to be constructed in computer memory - a process called "record". The computer "plays" the organ by invoking a note-table decode program which sets the organ latch bit corresponding to each note that is to be sounded. The organ, interface hardware, and record and playback software are described in Sections 3.3 to 3.6.

Since the pitches of notes and their start and end times are recorded independently, they are separately controllable when the organ is played back by the computer. Thus pitch transposition and playback speed may be separately altered, unlike a conventional tape-recorder. Because of the relatively high rate at which the keyboard is sampled, the computer's playing can incorporate subtle adjustments of tempo and nuances of performance. "Sound-on-sound" recording, analogous to tape recorder dubbing, is effected by "playing" from one note table while simultaneously "recording" on to another. The separate tables are subsequently merged. This feature has

proved to be very useful for recording complicated or difficult pieces of music.

These facilities can be invoked by typing the appropriate command mnemonic on the teletypewriter. Table 3.1 lists the commands which have been implemented. Response times for most commands are less than a second, and for those more complicated tasks where additional operator action is required, a short prompting message is automatically typed. The recorded note table may be displayed and edited in either TRAD or MOD notation. The TRAD display is discussed in detail in Chapter 4. The MOD notation is a positional notation similar to the one used, for example, by Brown (1959) in his composition "HODOGRAPH I". It is described in Section 3.7 and illustrated in Figure 3.5. Since the MOD display is isomorphic, in both a temporal and pitch sense, to the music it represents (unlike conventional notation with its essentially discrete, parametric representation of note durations) it is ideally suited to performance oriented tasks where nuances of performance or tempo must be depicted.

The MOD display is especially useful for "performance editing". This term refers to operations which alter the note table data and result in audible changes to the music when it is played back. In contrast, much of the TRAD display editing (cf. Chapter 4) is "typescript editing", which alters the presentation of the displayed symbols but does not introduce audible changes. Examples of performance editing are changes to note or chord start times and to pitches and durations. Examples of typescript editing are

alterations to stem directions or lengths, the insertion of beams, and annotation with text. TRAD display editing usually involves both performance and typescript editing procedures. However, the simplicity of the MOD display obviates the need for extensive typescript editing, while its temporal isomorphism to the sounded music permits subtle changes of note start times or durations to be made without difficulty. Phrases, chords or individual notes may be altered, deleted or inserted by suitably positioning the light spot with the joystick, typing the appropriate editing commands and specifying the correct notes either by teletypewriter commands or by depressing keys on the organ keyboard. Performance editing is described in Section 3.8, and the procedures required by the user are listed in Table 3.1.

Another useful approach to performance editing - editing in the aural rather than visual music domain - is inplemented using a foot switch in conjunction with playback or sound-on-sound recording. The foot switch (see Figure 3.1) activates an audible marker and permits the identification of individual chords or of whole sections of music. In addition, simple notational marks such as bar lines may be inserted. This facility is used mainly for deleting and inserting sections of music, where continuity of tempo is important. Most users find that the foot switch is inconvenient for specifying individual notes or chords; they prefer to use the joystick for this, presumably because it avoids manual timing problems.

A disc file facility permits edited note-tables to be

conveniently stored and rapidly retrieved. Each file incorporates an optional message feature, which permits the user to store a message or comments with each file. These comments are typed by the computer each time the file is retrieved. File retrieval overwrites the existing note-table. However, two additional commands have been planned to permit juxtaposition and concatenation of a filed note table with the existing note table. Medium term note table storage is provided by the magnetic tape disc-copy storage system described by Jordan (1974). High speed paper tape input/output provides long-term storage analogous to the disc file system.

The software design is modular, and is implemented as a multi-phase disc-based system with COMMON core data storage. A master interpreter together with phase overlay and executive modules provides overall control. A hierarchy of sub-controllers, with interpretive facilities where appropriate, is responsible for the execution of specific tasks. Both FORTRAN IV and EAI 640 ASSEMBLY languages are used, to effect a compromise between programming effort and machine independence, and storage efficiency and execution speed. The comparatively small core memory available (16K words) proved to be the major constraint affecting the software design. Thus, the data bases (Sections 3.4, 4.3) are very compact, and many of the display and editing algorithms favour storage economies in preference to execution speed. This consideration also led to the development of special I/O routines for the teletype, high-speed paper tape punch and reader, and display

oscilloscope. Software design and the system organisation are discussed further in Section 3.9.

From the user viewpoint, the system is flexible, convenient to use and rapid in response. The command mnemonics (cf. Table 3.1) are simple in form and permit substantial freedom of format. Each command consists of a single letter mnemonic which indicates the required task type, followed where appropriate by simple letter or numeric arguments which are delineated by commas. Both task type and argument mnemonics may be optionally spelt in full or in an abbreviated form. For example, the user-entered command "R↓" or "RECORD↓" or "REC↓" will initiate the record task. The character "↓" denotes the carriage return key, which indicates the end of the command. This flexibility has been found useful for people who are learning to use the system - since the command task names follow normal useage, it is natural for inexperienced users to enter the full task name. As more experience is gained the user desires faster response, and types only the command and argument mnemonic letters. Numeric arguments which may take integral or fractional values may be entered either as integers or as decimal numbers, without format restrictions. For example a change in playback speed ("velocity") may be entered as "V,2↓" or "V,2.0↓" (which results in a doubling of playback speed) or as "V,0.5↓" or "V,.5↓" (which results in a halving of playback speed).

Illegal command sequences as well as illegal commands are recognised by the interpreter. For example the "playback", "display" or "edit" tasks are illegal if a

"record", "get file" or "read paper tape" task has not yet been performed. A number of global system options are provided to permit flexibility without requiring excessive repetition or complication in the command arguments. These options are initialised to standard (default) values, and may be altered at any time using the command "X↓".

A typical session proceeds in the following manner. The musician/composer/student plugs in the organ keyboard and sound unit and loads the system magnetic tape on to disc using the EAI 640 Monitor and magnetic tape utility routines. Any one of the system phases - namely one of the core image files PIANO1, PIANO2, PIANO3 or PIANO4 - are then loaded into core and executed. The system initialises all data areas and options, and displays on the screen a brief explanation of how the system is operated, if this is desired. A list of available options is next displayed, with the current options indicated by an asterisk (see Figure 3.2). Option settings can be altered by typing the desired option code number - for example typing "2B↓" (see Figure 3.2(a)) will change the display notation option from MOD to TRAD. Since MOD and TRAD notations each require a different set of options, the options applicable at any time depend upon whether MOD or TRAD display option is requested. This is illustrated in parts (a) and (b) of Figure 3.2. Once this initialisation stage is completed, the computer types "←" to indicate that it is ready to accept commands. To record music from the organ keyboard, the user types "R↓" (or "RECORD↓") and plays the music passage on the organ. The record session is terminated when the next command is

typed. To display the music using the MOD notation option, the user types "S↓" or "SCORE↓". The procedure for the TRAD display is described in Chapter 4. Subsequent pages are displayed using the command "N↓" (or "NEXT↓") or "S,n↓" where n is the desired page number. The message: "CLOT! NON-EXISTENT PAGE" is typed by the computer if the requested page number is invalid. Typical display times for a full page are 2 to 3 seconds. To play back the recorded music, the user types "P↓". The command "P,n↓" permits playback starting from the beginning of page n. Playback is terminated when the next command is typed or when the entire piece of music has been played. Playback speed is altered using the command "V,n↓" where n is the integer or decimal factor by which the speed is to be multiplied. The MOD display time scale is not affected by this command - if the display time scale is also required to be modified then the additional command "V,F↓" must be used. To edit a note or chord on the currently displayed page the user positions the joystick so that the light spot (joystick cursor) coincides with the left side of the offending note symbol, and types and appropriate command. For example to delete a note the user points to the note symbol and types "D↓". To change a note pitch the user points to the note, types the command "C,P↓", and specifies the new note pitch by playing the required note on the organ keyboard.

Full details are given in the user manual, which is lodged in the Computer Laboratory of the Electrical Engineering Department.

3.3 ELECTRONIC ORGAN INTERFACE

The 1972 undergraduate project (see Section 3.1) used an analogue keyboard encoding scheme to interface the modified piano keyboard to the EAI 590. An analogue rather than digital approach was used because digital input facilities were not available, whereas a high resolution analogue interface forms an integral part of the Hybrid computer. Noise problems encountered with the analogue encoding hardware (Baird, 1972), together with the anticipated change from piano to organ, prompted the subsequent development of a 48 bit digital interface for both input and output of binary data. This I/O facility consists of three input and three output device controllers, each handling a 16 bit word transfer to or from the EAI 640 I/O bus under CPU command. Output bits are stored in latch memories incorporated in the output controller. RTL logic is used, primarily to permit direct interconnection with an existing RTL buffer between the actual 640 I/O bus (which uses CTμL logic) and the accessible bus terminals. This buffer was developed by Cashin and Mayson (1969) to facilitate the development of special peripheral devices within the Department, while providing isolation and protection for the 640 hardware.

The organ unit used in conjunction with the RTL I/O Interface is a modified "PIANOMATE" - a commercially available 4 octave instrument intended to supplement a piano. It uses 16 individual discrete-component oscillators, with each oscillator shared between three adjacent notes by

switched resistors. While this oscillator grouping technique reduces the hardware cost and power consumption, it inhibits flexibility since if more than one note in any group is played only the lowest pitched note of the played notes is sounded. This restriction led us to incorporate a further 32 oscillators so that each note can be sounded independently.

A useful feature of the PIANOMATE is its keyboard switches, which are portable. Two banks of 24 switches sit on any piano or organ keyboard to cover 48 keys. Each switch is connected to its key by a lightweight "finger", which drops when the key is depressed and thereby activates the switch. We use these switches in conjunction with a portable soundless practice keyboard.

In the original PIANOMATE unit each key switch activates its oscillator by direct connection of a stable 25.0 V supply. This arrangement is unsuitable for our application because each switch must be sampled by the RTL I/O Interface, and because each oscillator must be activated by either a logic (latch) bit or by its key switch. Consequently each key switch generates directly a logic signal, and transistor switching is employed to translate logic level signals to the 25.0 V required to activate each oscillator. RTL positive logic conventions are used throughout.

A schematic of the RTL I/O Interface and Organ system is given in Figure 3.3, while Table 3.2 summarises the 640 CPU and I/O controller action and defines the signals indicated in Figure 3.3. Full documentation is held in the

Electrical Engineering Department, University of Canterbury.

Although the RTL I/O Interface was designed primarily to suit the requirements of the organ I/O system, it was sufficiently comprehensive to attract other users until the commissioning of the general-purpose 8 word BDI in early 1976 (cf. Section 6.6). In accordance with Departmental policy, the RTL I/O Interface has been removed and the organ is now interfaced using the BDI. To take advantage of the extended facilities of the BDI, a new 5 octave organ was constructed in conjunction with the digital synthesiser/organ (cf. Chapter 6) by Vaughan (1977). This organ uses a crystal-controlled master oscillator with digital frequency division to generate a square wave pulse train for each of its 60 notes.

To provide a consistent presentation, the 5 octave, 60 note organ is used in the software description which follows.

3.4 MOD DATA STRUCTURE

The data structure used for the record, playback, sound-on-sound and MOD display and edit tasks consists of three parameters for each note. These parameters are: note start time, pitch and duration; and are packed into two consecutive 16 bit words to conserve storage. Parameter packing and unpacking routines are provided to facilitate the reading and writing of table entries. The COMMON area used for the MOD note table is shown in Table 3.4.

Note start time, measured in samples (or equivalently in centiseconds because of the 100 Hz sample rate used), is

allocated 16 bits and occupies the first word of each entry. The time origin is aligned to the start time of the first note in the table, and all note entries occur in chronological order so that start times increase monotonically. For convenience, all start times are positive so that effectively only 15 bits are utilised. This convention places an upper limit of 5.46 minutes on start time values before overflow occurs. While this limit can be doubled by including negative values (i.e. by defining the time origin as $-(2^{16}-1)$ rather than 0) the restriction of table storage to 2000 words or 1000 notes is usually a more severe limitation.

Note pitches are encoded in 7 bits, using the code defined in Table 3.3. This corresponds to a labelling of the piano keyboard in pitch-ascending order. Since this representation is isomorphic to the keyboard, it is independent of key signature or of any modulations which occur.

Note durations are measured in samples, and encoded in 9 bits. This convention restricts the maximum duration permitted in any note entry to 511 samples, or 5.11 seconds at the standard sampling rate. This limitation is overcome by the use of multiple entries for notes exceeding this duration. Thus, when a note duration reaches 510 samples it is artificially terminated, and a new note entry started so that an overlap of several samples is provided. Utility routines are available to test for this condition and to calculate the effective total duration of any note, so that this "note breaking" effect is transparent. Since most

notes are less than 5 seconds long, the storage advantages of the packed data structure amply compensates for the additional control software complexity required.

Bar line markers are treated as a special note of pitch '177 and zero duration. These are used in the TRAD transcription and display procedures described in Chapter 4, and are inserted using the MOD edit facilities discussed in Section 3.8.

Figure 3.4 shows details of the MOD data structure bit utilisation, and illustrates the sequential rather than linked list nature of the table. This sequential structure provides significant economies in both storage and access time in comparison with a linked list structure (cf. Fredlund and Sampson, 1973). These advantages outweigh the inconvenience of having to shift substantial portions of the table when entries are inserted or deleted during editing - a disadvantage avoided by the linked list structure. However, the effect of this on edit task execution time is only small because of the use of fast ASSEMBLY language edit modules which efficiently perform the necessary table reorganisation.

The data structure described above is similar to those used elsewhere. Knowlton (1971) uses a 20 Hz keyboard sampling rate and employs an incremental rather than absolute approach to start time measurement. Thus Knowlton stores the time interval between successive note start times. While this approach avoids the time measurement overflow problem mentioned above, it increases the complexity of some edit operations. Longuet-Higgins (1976) describes a

structure identical in concept to ours, but does not utilise parameter packing for storage efficiency.

3.5 RECORD SOFTWARE

The record software periodically samples the entire organ keyboard while it is being played, and constructs a table of note parameters in the MOD data format described in Section 3.4. This latter process is essentially an information transformation which severely reduces the redundancy inherent in the keyboard samples. A useful feature which is incorporated is software protection against keyboard switch noise and switchbounce. ASSEMBLY language is used for speed and because of the convenience it possesses for bit manipulation tasks.

To achieve flexibility and to permit the sharing of software between the record and sound-on-sound tasks, the record process is divided into two software modules. The first (subroutine RECORD) initialises pointers and data areas, enables the 100 Hz interrupt clock which defines each keyboard sample instant, and samples the keyboard switches. Each keyboard sample interrupt causes the sequential input of four data words, each of 16 bits. Since successive data word transfers occur in a time interval which is much shorter than the interval between successive keyboard samples (approx. 10 μs compared with 10 ms), all 60 bits can be regarded as being transferred in parallel, and keyboard sampling jitter can be neglected. Following the task initialisation, the keyboard is thus monitored until the first key is depressed. At this sample instant the time

origin is established and the record process formally commences.

The four keyboard sample words are transferred to the second module (subroutine REC) which sequentially compares each bit with the corresponding bit of the previous sample to ascertain the current note status. The possible bit configurations and the corresponding note status and program action are summarised below:

Sample Bit Configuration and Note Status

Previous Sample	Current Sample	Note State	Action
0	0	Note not played	Look at next note bit.
0	1	New note starts now	Initiate duration counter, start new table entry (store start time).
1	1	Note continues	Update duration counter.
1	0	Note ends now	Terminate note entry (Store duration and pitch).

When all 60 bits have been processed, REC returns control to RECORD which pauses until the next sample interrupt occurs. Bit processing is sufficiently rapid that about 90% of the total record processing time is spent in this pause.

Since record is an open-ended task (in the sense that it has no intrinsic termination condition), it proceeds until the next command is specified on the teletype. Thus once every sample interval RECORD tests the teletype status for a pending command, and if present accepts it, terminates the record task, inhibits the sample clock interrupts, and completes all note-table entries and associated parameters.

Control is then transferred to the master interpreter for execution of the specified command.

Switchbounce and noise protection is implemented within REC by delaying the commencement and completion of note-table entries by several keyboard samples, until a valid note start or end is confirmed. Thus, a note start indication is assumed to be due to noise until its duration exceeds 5 samples. Similarly, a note end indication is assumed to be due to switchbounce until it has been "off" for 5 samples. If a switchbounce condition is detected, the duration counter is updated to compensate for the "missed" samples. This technique is more elaborate than a final note-table tidying operation (e.g. to delete all notes of duration less than 5 samples, or to bridge consecutive notes of the same pitch whose end and start times are closer than 5 samples), and requires a small intermediate storage buffer. However, if a final note-table tidying approach is used then each note which is affected by noise and/or switchbounce will require multiple note-table entries during the record process. Thus the effective storage area is reduced and premature table overflow may occur.

3.6 PLAYBACK SOFTWARE

The ASSEMBLY language playback software reads note start time, pitch and duration parameters stored in the MOD format data table and generates the successive 60 bit keyboard "samples" required to duplicate the keyboard performance. These keyboard samples are transferred to the organ latch bits to sound the notes currently played.

This procedure, which is the inverse of the record process, is not as elegant as selectively setting or resetting at the appropriate instants only those notes which start or stop, but is dictated by the simple 4 word serial, 16 bit parallel hardware latch output system used (see Section 3.3).

An interval timer interrupt is used to initiate the processing of successive "sample" instants. The interval timer facility, developed under the supervision of W.K. Kennedy, permits software control of the interrupt rate. Thus playback speed can be altered by suitable adjustments to the "sample" rate, as well as by explicit transformations of the note table time values. Speed-up factors of up to 10 are possible using the former technique before the processing time required for each "sample" becomes critical. The "V,n" and "V,F" commands (see Table 3.1) are used to specify which of these two speed control methods is used.

The technique used to generate the bit patterns for each keyboard "sample" is as follows. A 60 word integer "duration" array, which is indexed by pitch, is initialised to zero. A software counter is incremented each "sample" instant to provide a time count, and is used to establish the "sample" at which each note starts. When a note starts, the array word corresponding to its pitch is set equal to the negated duration value of the note. Thus, all notes which are played at any "sample" instant may be identified by the sign bits of the array words. This permits the rapid generation of the keyboard bit pattern. Once this bit pattern is generated for each sample it is transferred to

the organ latch bits and all negative words in the duration array are incremented. The requirement that all MOD data entries in the note table occur in chronological order (cf. Section 3.4) simplifies the procedure which recognises the "note starts now" condition.

For convenience to the user, playback is terminated prematurely when the next command is entered on the teletype. This facility, together with the "P,n" command which specifies that playback commences at the start of page n, gives the user considerable freedom in specifying the passages desired for playback.

A pitch transposition feature which permits music originally recorded in one key to be played back in a different key is implemented in two ways. The command "T,+n" or "T,-n" (see Table 3.1) causes each note to sound n semitones above (or below) the original pitch, but leaves the stored note table unaltered. A global pitch transposition factor which is set to ±n modifies the pitch of each note entry as the latter is read. The command "T,F" causes all stored pitch values to be modified and the transposition factor to be reset to zero. An additional command ("T,R") is provided to reset the transposition factor without modification of the note-table entries. To avoid confusion, successive applications of the "T,±n" command modify the transposition factor relative to its current value, using algebraic addition. These transposition facilities apply also to the MOD display facility discussed in Section 3.7.

The sound-on-sound feature permits the recording of

successive "layers" of music, by playing from one note table while simultaneously recording on to another. The playback controller performs the data initialisation and keyboard sampling normally done by RECORD, while REC is used to generate the additional note table from the keyboard samples. In this manner the playback and record sampling intervals are synchronised. The sound-on-sound task, like record, is terminated by the next teletype command. The new note table is then transposed by minus the current pitch transposition factor (to maintain subsequent parity between tables), and the two tables are merged by juxtapositioning and sorting. The simple "bubble-sort" algorithm described by Page and Wilson (1973) is used to ensure that the requirement that note-start times increase monotonically is satisfied. While this sorting procedure can take up to 30 seconds under extreme conditions, it is normally completed within several seconds. The use of more sophisticated sorting algorithms such as the distributive methods "quicksort" and "monkey puzzle sort" (cf. Page and Wilson, 1973) is not considered to be justified, especially in view of the need for storage efficiency.

The foot switch facility, described in Section 3.8, is used in conjunction with playback or sound-on-sound recording to identify time instants. This is achieved by testing the status of the foot switch once every sample instant.

3.7 MOD DISPLAY

The MOD display facility provides a visual display of the stored note table in a graphical pitch/time notation which is isomorphic to the note parameters. Figure 3.5 illustrates the notation and demonstrates several optional features. Notes are represented by blocks on a pitch/time continuum, with time and pitch as the horizontal and vertical ordinates, respectively. A conventional piano stave set provides a pitch reference, and conventional notation pitch conventions are followed with the exception that black keyboard notes are positioned vertically midway between the adjacent white notes, and no account is taken of key signature or modulations. Since this pitch representation can lead to confusion between black and white note pitches because of the relatively small vertical displacements used, an optional feature is provided whereby black and white keyboard notes are represented by hollow and solid blocks respectively (Figure 3.5(b)). A further option permits control over the block widths (Figure 3.5(c) and (d)). Since time representation is continuous rather than discrete or parametric (cf. conventional music notation, discussed in Chapter 4), no general provision can be made for justification (using this term in its typesetting sense). Thus, a note may extend past the end of a stave group. In this case it is "broken" at the stave group end, and continued on the next stave group. Initially, an asterisk marker symbol was used to identify such broken notes, but this convention has been abandoned as it was found to

confuse the display.

Bar line markers (Section 3.4), if present, may be suppressed or displayed as a dashed vertical line. Text insertion and deletion facilities, described in Chapter 4, permit annotation of the display and the addition of accent marks.

Changes to the effective time scale are achieved using the "V,n" followed by "V,F" commands, which transform the note-table time parameters by the factor 1/n (cf. Section 3.6; Table 3.1). A facility to permit alterations to the display time scale independently of the note table was considered but not implemented. The display scale is standardised at one centisecond per horizontal resolution element, or about one second per 2.0 cm. Thus the effective length of a stave group is 7.36 seconds.

The use of a storage rather than refresh display oscilloscope dictated the use of a fixed page display, rather than the moving scroll display used by Ashton (1970). This led us to abandon the idea of implementing a real-time display which is generated simultaneously with the record task, despite the technical feasibility of so doing. Since most display and edit applications require the fast display and random page access facilities provided, the development of an additional real-time display mode was not considered to be justified.

The display software, which is in ASSEMBLY language, is organised in three modules. These control the drawing of an entire page, the generation of symbol placement and length parameters, and the drawing of note block symbols.

The concept of an "invisible display", whereby note display parameters are generated but symbol drawing is suppressed, permits the use of the display software for other tasks and is implemented using global steering flags. Thus, a particular note which is pointed to by the joystick for some edit operation (Section 3.8) is located by comparing its display co-ordinates with the sampled joystick co-ordinates. Similarly the first note on a given page is identified by the sequential "invisible" display of previous pages, so that random access to individual pages is permitted for display or playback (cf. "S,n" and "P,n" commands, Table 3.1). This latter facility requires the invisible display rather than a stored table of pointers, because the time co-ordinate of the page start is not in general equal to the start time of the first note, and because "broken" notes which continue from the previous page may exist.

The display control module (subroutine SCORE) is responsible for setting the display mode flags and initialising new page parameters, for locating the first note on the required page (using subroutine PAGENM), and for generating the display stave lines and numbering the page (using subroutine STVDW1). The note display procedure is controlled by subroutine XYGENM, which generates for each note in turn the co-ordinates and length of the note block symbol. Each block symbol is drawn by subroutine NTDRWM which uses the vector drawing software described in Section 4.5.4, and which takes into account the block width and hollow/solid symbol options mentioned above. The parameters of "broken" note blocks are stored in a small

circular buffer by XYGENM, and at the beginning of each new stave group the continuing note blocks are drawn. Bar line markers, if present, are decoded and drawn as dashed vertical lines by NTDRWM.

Typical page display times are 2 to 3 seconds.

3.8 EDITING FACILITIES

Comprehensive edit facilities permit individual notes, chords or entire sections of music to be altered, inserted or deleted. This editing may be effected using the MOD display and joystick or playback and foot switch to specify the notes, chords or sections which are to be changed. As discussed in Section 3.2 such edit processes are called "performance editing" because they are manifest in the corresponding sounded music.

Table 3.1 lists the edit facilities available and the required user action. Edit task mnemonics which use the display joystick are "C" (change), "D" (delete) and "I" (insert), while the subsequent alphanumeric argument(s) specify the parameter(s) which are to be changed. When an edit task is recognised by the master interpreter the edit control interpreter (subroutine EDIT) is called and the joystick cursor co-ordinates are read. Subroutine EDIT then decodes the argument(s) and supervises the specified task. When the task is completed the computer types "←" and accepts and interprets the next user command. Non-edit commands cause control to return to the master interpreter. If an error condition is encoutered, for example a note is

not located at the specified joystick position, the computer types "?" together with a short error message such as "NOTE NOT LOCATED". The current task is then aborted and the next command is accepted. To simplify the command mnemonics, operations on notes and chords are specified using the same command. When the note currently pointed to is located, it is tested for chord membership using the criterion that another note must exist such that the difference in start times is less than 5 samples. If chord membership is detected, the user specifies whether the current operation refers to the note specified or to the entire chord, by responding "Y" or "N" to the request "NOTE OR CHORD?". This concept of hierarchal levels of command interpretation, with operator response at appropriate points, is used extensively throughout the system. It has been found to simplify the command structure and interpretive software as well as increasing the flexibility and convenience to the operator.

Most edit operations conveniently subdivide into distinct modules which are common to several tasks. This intrinsic modularity has been utilised so that at the control level the execution of an entire edit command reduces to a sequence of subroutine calls. Thus, modules are available to read the joystick co-ordinates, locate the note pointed to using the EDIT DISPLAY mode (cf. Section 3.7), test for chord membership, delete individual notes comprising possibly multiple data entries (cf. Section 3.4), delete individual data entries, specify new note pitch(es) by sampling the organ keyboard, insert new notes, etc.

Further detail here is not warranted - the interested reader is referred to the software listings which are extensively documented. Copies are held in the Computer Laboratory of the Electrical Engineering Department.

Insertions of notes, chords or sections are handled by setting up the new note entries in a separate data table, and invoking the juxtapositioning and sorting modules used in the sound-on-sound task (Section 3.6). This approach is also used in the change pitch command ("C,P") - thus the existing note(s) are deleted, the new pitch(es) entered via the organ keyboard, and the corresponding new note(s) inserted. This method permits a note or chord to be replaced by a different chord, so that the number of notes as well as pitches are altered.

Operations on sections use time instants rather than note pointers to specify the section parameters. This permits complete generality, because note portions or arbitrary time sections can be inserted or deleted. Time parameters are calculated using the temporal isomorphism between the sounded music and corresponding display, or using the footswitch in conjunction with playback. Two options exist within the "delete section" task - the section time interval can be completely deleted and the start times of all subsequent notes adjusted by subtracting the section duration, or the notes within the section can be deleted, leaving a silent interval. These options are specified by the operator response "Y" or "N" to the request "CLOSE THE GAP AFTER DELETING SECTION?". A sound-on-sound section insertion facility which creates a time interval complements the straightforward sound-on-sound facilities. The "O,I"

(or alternatively "P,I") command causes playback to occur until the foot switch is first depressed, or the first note played manually on the organ keyboard. The time instant thus specified defines the start of the new section, and playback ceases. The record task immediately commences, and proceeds until the foot switch is next operated. Playback then re-commences. Before the new and old note tables are merged, the original table is modified by retarding the start times of all entries which commence later than the inserted section start time.

The insertion of bar line markers, which are required for the TRAD display transcription (cf. Chapter 4), are inserted either using the "I,BL" command with the joystick and display or using the "B" command with the foot switch and playback. They may also be entered using the foot switch during the record session, but this method is not used as widely as expected because most pianists experience difficulty in accurately synchronising their feet and fingers. Of the available insertion methods, the joystick/ display method is the most widely used. For user convenience, this method incorporates a multiple bar line entry facility. Thus, the first marker is inserted at the joystick position when "I,BL" is typed, and subsequent markers are inserted each time the "RETURN" key is depressed. This multiple bar line insertion facility terminates when the next command is entered.

The edit facilities discussed above parallel or extend those available with a conventional tape recorder, but with considerably enhanced convenience and control.

Thus, time resolutions down to a centisecond are possible. Response times for all commands are of the order of a second, and rapid checks on the resultant note table may be performed visually or aurally. Time reversal has not been implemented for general use because of a lack of demand - it is incorporated as a retrograde permutation with Susan D. Frykberg's composition aid (cf. Section 3.1). Facilities for concatenating or juxtapositioning individual sections have not been developed because this can be done using sound-on-sound. In addition, juxtaposition of individual sections may be unsatisfactory for the user because of the need to specify registration markers, and because of inevitable tempo differences.

The preceding comments are not intended to imply that the Piano Typewriter system can universally replace a tape recorder. However, for those applications where note timbres and pitches can be generated or controlled separately from note temporal characteristics then the comments of the previous paragraph apply. This is particularly pertinent to the development of computer-controlled music synthesisers.

3.9 SOFTWARE ORGANISATION

Core storage restrictions dictate the use of a multi-phase system which employs successive phase overlays by special executive routines in conjunction with the disc operating system. The design of this phase overlay system is simplified because both data and global system parameters may be retained in core in COMMON data area. Thus only

executable programs in core image form need be loaded from disc. This approach has resulted in global storage conventions which are summarised in Table 3.4. About 5K of the total available 16K words core storage is used for data and system variables. Table 3.5 summarises the contents and function of each core-image phase.

Phase overlays are controlled by the master interpreter, through a small mainline routine which occupies approximately '230 words and which is common to all phases. Modification of these two programs to accommodate new phases is straightforward, so that the entire system can be conveniently and rapidly interfaced to other programs which have been developed separately. Thus, the commands "M", "H" or "L", which refer to tasks in Lamb's (1977) music analysis and teaching system, cause that system to be loaded into core and executed. Similarly commands prefixed by "." and "#" cause control to be transferred to Miss Frykberg's composition aid or Vaughan's (1977) digital synthesis system, respectively. These systems are programmed to return to the master interpreter when the appropriate command is entered by the user. In this way the transition between the various systems is transparent to the user. The flexibility of the master control interpreter and phase overlay executive system permits the Piano Typewriter to be regarded as a general-purpose operating system for music oriented programs.

Some programming difficulty is experienced when nested subroutine calls require intermediate phase overlays, since this causes the subroutine return address to be destroyed. With ASSEMBLY language modules, return addresses

are easily accessible and may be saved in a COMMON area for subsequent restoration. This approach is not convenient with FORTRAN programs. In consequence many programs are designed with multiple entry points so that a return is effected by a CALL ... rather than a RETURN statement, and the return address is not required.

The master interpreter uses both STATE and OPTION parameters. OPTION parameters are altered by the "X" command (Table 3.1) and are used as control steering variables (e.g. MOD or TRAD display option). STATE variables are used for internal housekeeping operations, such as keeping track of which phase is currently operational, or testing for illegal command sequences.

3.10 CONCLUSIONS

The interactive Piano Typewriter system described in this chapter provides a useful interface between a musician/composer/teacher/student and a digital computer. It permits sounded music to be conveniently entered into and played back by the computer using an electronic organ, provides flexible display and edit facilities, and incorporates efficient storage of the music. It is oriented to the needs of musicians, composers, teachers and students by utilising media with which they are already (or can rapidly become) familiar. This permits them to approach the system with a sense of convenience. Two complementary display notations - MOD and TRAD (this latter is the subject of Chapter 4) - enhance this convenience.

The aim throughout the development of the system has been to complement rather than supplant the creative human, by providing useful musical tools. This concept is reflected in the terminology "piano typewriter", and in the fact that the computer on which the system is implemented is essentially transparent to the user. Nevertheless the "music operating system" design does permit, if so desired, the incorporation into the system of computer programs developed specially by the user. The work by Lamb (1977), Susan D. Frykberg, and Vaughan (1977) has been included in this manner to provide additional facilities.

The system is used for teaching music theory and keyboard practice, administering aural tests, composing music, and generating traditionally notated music scores (Tucker *et al.*, 1977). The comments and reactions of the composers, teachers and musicians who have seen or used the system have been very encouraging, and many of their suggestions have been incorporated.

TABLE 3.1

SUMMARY OF COMMANDS

Where subsequent action is required by the operator, a short prompting message is typed.

PART A	MOD DISPLAY OPTION
R	Record music played on organ keyboard, until next command is entered.
P	Playback, starting from beginning. Terminate when end of piece is reached, or when next command is entered.
P,n	Playback, starting from page n. (n a positive integer).
V,n or V,r.s or V,.s	Speed up playback by factor n or r.s or .s (note-table is unaltered). (n a positive integer, r.s and .s are positive decimal numbers).
V,R	Reset playback speed to original value.
V,F	Fix playback speed by modifying note-table time values. Set playback speed factor to standard value.
O	Sound on sound (Overlay). Playback is preceded by a 3 sec. lead-in, with 3 "beeps" at 1 sec. intervals. Simultaneously, record from organ keyboard. Terminate when next command is entered.
O,n	Overlay, but start playback from page n.
S	Display score in MOD notation, starting at page 1.
S,n	Display page n in MOD notation.
N	Display next page.
T,+n or T,-n	Transpose by +(-)n semitones, relative to present transposition factor. Note-table pitch values are not altered. (Transpose commands apply to both display and playback.)
T,F	Fix transposition, by modifying note-table pitch values and setting transposition factor to zero.
T,R	Reset transposition factor to zero.

TABLE 3.1 (Continued 2)

X	Change options. Current options are displayed, together with a list of all available options.
F,NAMEXX	Create a disc file named NAMEXX and store all note-table and text-table data, as well as existing options.
G,NAMEXX	Get disc file named NAMEXX and overwrite all existing note-table and text-table data and existing options.
$W	Create (write) a paper-tape dump of all data, as for disc file facility.
$R	Read a paper tape of all data, as for disc file facility.
$L	List data type (MOD, TRAD) and data table on teletype.
H	Transfer control to Harmonise module controller. (M.R. Lamb's system - see Section 3.9).
M	Transfer to Mark module controller. (M.R. Lamb's system - see Section 3.9).
L	Transfer to Learn module controller. (M.R. Lamb's system - see Section 3.9).
.c	Transfer to Interactive Inspiration module controller. (c is any valid command handled by the I.I. Controller). (S.D. Frykberg's system - see Section 3.9).
#c	Transfer to Digital Synthesiser-Organ module controller. (c is any valid command handled by D.S.O. controller). (R.G. Vaughan's system - see Section 3.9).
B	Delete all existing bar-lines, and insert new bar lines using sound-on-sound facility.
E	Exit. (Transfer control to Monitor operating system).
A	Roundoff. As for TRAD roundoff (note start times and durations are aligned), but MOD data format is retained.

TABLE 3.1 (Continued 3)

MOD EDIT FACILITIES USING PLAYBACK AND FOOTSWITCH

P,I or O,I	Insert section using playback with overlay. New section starts when footswitch is first pressed, or when first note is played on organ keyboard. New section ends when footswitch is next pressed. Playback ceases during new section insertion and recommences where it stopped when new section ends.
P,D or O,D	Delete section using playback. Deleted section start and end times are defined by footswitch operation. Time gap corresponding to deleted section may be kept or deleted.

MOD EDIT FACILITIES USING DISPLAY AND JOYSTICK

D	Delete note or chord. (Start of note is pointed to).
D,S	Delete section. (Section start is pointed to). (Section end is next pointed to and "E" typed). Time gap corresponding to deleted section may be kept or deleted.
D,B	Delete bar. (Anywhere in bar is pointed to).
D,BL	Delete all bar lines.
D,L	Delete all little notes. (Smallest valid note duration is entered next).
D,O	Delete the shortest of all overlapping notes (i.e. leave "melody").
D,P/	Delete section from start of page to "here" (specified by joystick).
D,/P	Delete section from "here" to end of page.
D,P	Delete section from start of page to end of page.
D,/	Delete from "here" to end of piece.
D,T	Delete text message. (First character in message is pointed to).

TABLE 3.1 (Continued 4)

C,P	Change pitch of note or chord. (Start of note, or a note in the chord, is pointed to). Play new note or chord on organ keyboard - a half second time interval from start of first note played to sample instant is allowed, to compensate for note start time variations in new chord. Number of notes in new chord may be different to number in original note or chord. Note start time and duration of note pointed to are used in new note or chord entries.
C,S	Change start time of note or chord. (Start of existing note is pointed to). (New start time is next pointed to, and "S" typed).
C,D	Change duration of note or chord. (Start of note is pointed to). (New end time is next pointed to, and "E" typed).
I	Insert a note or chord. (New note start is pointed to). New end time is next pointed to, and "E" typed, then note pitch(es) played on organ keyboard.
I,BL	Insert a bar line. (Bar line X position is pointed to). Subsequent bar lines are inserted each time the "RETURN" key is depressed, until the next command is entered.
I,T	Insert text. (First character position is pointed to). Character size is entered on teletype, then message is typed. Audible "beep" warning is given when end of line is nearly reached.

TABLE 3.1 (Continued 5)

PART B	TRAD DISPLAY OPTION
S	Display (Score) in TRAD. If data format is MOD, set up time and key signatures and check that bar lines exist. If no bar lines are present, get operator to insert them using B or I,BL commands. Perform roundoff and data format conversion to TRAD data form. When TRAD data form exists, display page 1 in TRAD notation.
J,n,m	Justification control. Line n on current page is required to have m bars in it. (n,m are positive integers).
D or D,N	Delete note. (Note head symbol is pointed to).
D,R	Delete rest. (Rest centre is pointed to).
D,A	Delete accidental. (Accidental symbol is pointed to).
D,B	Delete bar line. (Bar line top is pointed to).
D,T	Delete text message. (First character is pointed to).
D,L	Delete link (beam). (Any note in the beamed cluster is pointed to).
C,P	Change pitch of note or rest. (Note head or rest is pointed to). New pitch value is entered from teletype.
C,D	Change duration of note or rest. (Note head or rest is pointed to). New duration value is entered from teletype.
C,K	Change key signature. New key signature is entered from teletype.
C,T	Change time signature. New time signature is entered from teletype.
C,X	Change X co-ordinate of note or rest. Note head or rest is pointed to. (New symbol position is next pointed to, and "RETURN" key pressed).
C,S,U	Change specified note stem direction to up. Stem length is not changed.

TABLE 3.1 (Continued 6)

C,S,D	Change specified note stem direction to down. Stem length is not changed.
C,S,L	Change specified note stem length. (New stem end position is next pointed to, and "RETURN" key pressed). Simultaneous changes of stem length and direction are thus possible.
C,N,T	Change specified note type to treble. (Note pitch will be subsequently drawn relative to treble clef).
C,N,B	Change specified note type to bass. (Note pitch will be subsequently drawn relative to the bass clef).
C,N,R	Change note into rest. (Note head is pointed to).
C,4,S	Set to 4 part harmony type S. (Treble clef, tail up).
C,4,A	Set to 4 part harmony type A. (Treble clef, tail down).
C,4,T	Set to 4 part harmony type T. (Bass clef, tail up).
C,4,B	Set to 4 part harmony type B. (Bass clef, tail down).
I,T	Insert text. As for MOD command.
I,L	Insert link (beam). First note in cluster is pointed to. Middle and end notes are subsequently pointed to.
I,U	Insert a unison note. (Lonely note is specified).
I,N	Insert a note. A reference note is pointed to. New note is inserted in same chord as reference note, or a new chord is started to left or right of reference note. Pitch and duration are specified from the teletype.
I,R	Insert a rest. Same as for I,N but rest is inserted.
I,A	Insert accidental. Note head is pointed to. Accidental type is specified from teletype.

TABLE 3.2

640 AND I/O CONTROLLER ACTION
SEE ALSO FIGURE 3.3

INPUT

On execution of instruction DI DEVNO, the device number DEVNO (bits 8-15 of Instruction Word) is put on to the Address Lines, and the DIL is raised. Each device compares the Address Lines with its device number. The device finding agreement enables information on to the data bus, and then raises its DRL. The 640 samples the data lines and lowers the DIL. The device then lowers the DRL, and the 640 continues program execution.

The controller action is summarised by:

ADROK	=	1 IFF ADR = DEVNO	(Address Decoded OK)
DIOK	=	ADROK • DIL	(Controller recognises input required, this device)
DTI(I)	=	K(I) • DIOK	(Data for CPU is gated on to bus)
DRL	=	DIOK	(Device flags CPU)

OUTPUT

On execution of instruction DO DEVNO, the device number DEVNO is put on to the Address Lines, and the data in the A register is put on to the Data Lines. The DOL is then raised. Each device compares the address lines with its device number. The addressed device samples the Data Lines, and then raises the DRL. The 640 lowers the DOL, the device lowers the DRL, and the 640 resumes program execution.

ADROK	=	1 IFF ADR = DEVNO	(Address Decoded OK)
DOOK	=	ADROK • DOL	(Controller recognises output required, this device)
N'(I)	=	DTO(I) • DOOK	(Data from bus is accepted)
DRL	=	DOOK	(Device flags CPU)

TABLE 3.3

PITCH ENCODING TABLE

Note Pitch	Pitch Code	Note Pitch	Pitch Code
A0	1	F4	45
A#0	2	F#4	46
B0	3	G4	47
C1	4	G#4	48
C#1	5	A4	49
D1	6	A#4	50
D#1	7	B4	51
E1	8	C5	52
F1	9	C#5	53
F#1	10	D5	54
G1	11	D#5	55
G#1	12	E5	56
A1	13	F5	57
A#1	14	F#5	58
B1	15	G5	59
C2	16	G#5	60
C#2	17	A5	61
D2	18	A#5	62
D#2	19	B5	63
E2	20	C6	64
F2	21	C#6	65
F#2	22	D6	66
G2	23	D#6	67
G#2	24	E6	68
A2	25	F6	69
A#2	26	F#6	70
B2	27	G6	71
C3	28	G#6	72
C#3	29	A6	73
D3	30	A#6	74
D#3	31	B6	75
E3	32	C7	76
F3	33	C#7	77
F#3	34	D7	78
G3	35	D#7	79
G#3	36	E7	80
A3	37	F7	81
A#3	38	F#7	82
B3	39	G7	83
C4	40	G#7	84
C#4	41	A7	85
D4	42	A#7	86
D#4	43	B7	87
E4	44	C8	88

TABLE 3.4

COMMON AND ZONE-ZERO STORAGE CONVENTIONS

AREA NAME	LENGTH (WORDS)	USE
ZONE-ZERO	128	GLOBAL SYSTEM FLAGS
TABLE/LIST1	2000	MOD NOTE TABLE SECONDARY NOTE TABLE (TRAD)
TABLE/LIST 2	2000	SOUND-ON-SOUND TABLE (MOD) PRIMARY NOTE TABLE (TRAD)
SCRTCH/DUMMY	200	MISCELLANEOUS BUFFER
YODEL/VOICE	200	MISCELLANEOUS (MOD) NOTE LINK BUFFER (TRAD)
PAGET/NPAGE	80	MISCELLANEOUS (MOD) PAGINATION TABLE (TRAD)
ORDER/ANS	80	TELETYPE INPUT
TTATR/ITEXT	160	DISPLAY TEXT TABLE
STATEV/STATE	5	STATE VARIABLES
STATEV/OPTION	10	OPTION VARIABLES

FURTHER DETAILS ARE GIVEN IN THE SOFTWARE DOCUMENTATION

TABLE 3.5

CORE-IMAGE PHASES - FUNCTIONAL LAYOUT

PHASE NAME	FUNCTIONAL MODULES PRESENT
PIANO1	MASTER INTERPRETER RECORD PLAYBACK SOUND ON SOUND EDITING USING FOOTSWITCH SET TRANSPOSITION FACTOR, PLAYBACK SPEED
PIANO2	MASTER INTERPRETER MOD DISPLAY MOD EDIT
PIANO3	MASTER INTERPRETER CONVERSION FROM MOD TO TRAD DATA BASE (CHAPTER 4)
PIANO4	MASTER INTERPRETER DISPLAY AND SET OPTIONS DISC FILE I/O
TRADT1	PAPER TAPE I/O (MOD AND TRAD)
TRADT2	(SEE CHAPTER 4) TRAD INTERPRETER TRAD DISPLAY CONTROL TRAD JUSTIFICATION TRAD EDITING
TRADT3	(SEE CHAPTER 4) TRAD SYMBOL DISPLAY

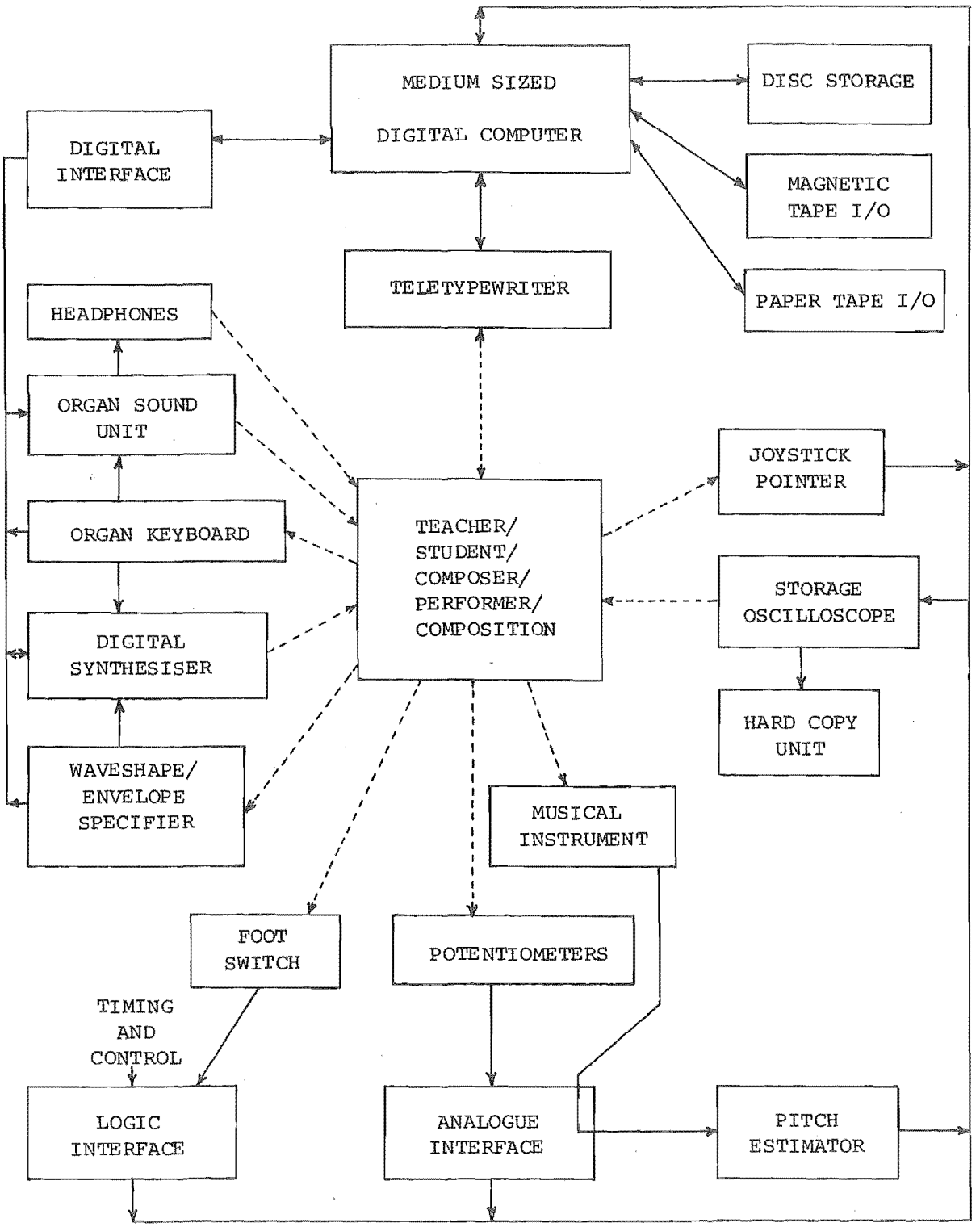


Figure 3.1 Block diagram of the system.

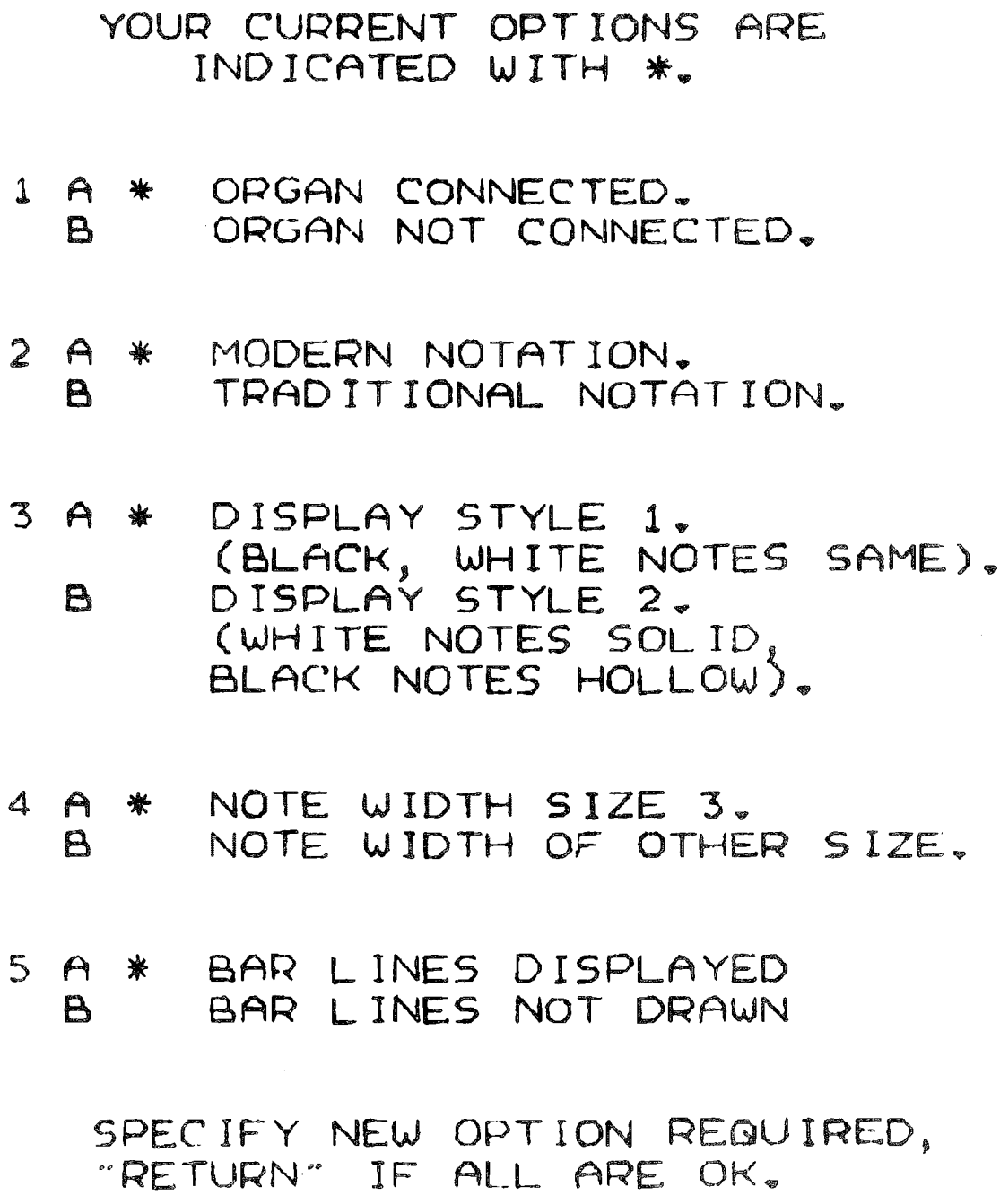


Figure 3.2(a) Display showing the options available, MOD notation selected.

YOUR CURRENT OPTIONS ARE
INDICATED WITH *.

1 A * ORGAN CONNECTED.
B ORGAN NOT CONNECTED.

2 A MODERN NOTATION.
B * TRADITIONAL NOTATION.

3 A DISPLAY FORMAT 1.
(SINGLE VOICE PER CLEF).
B DISPLAY FORMAT 2.
(PIANO MUSIC).
C * DISPLAY FORMAT 3.
(4 PART HARMONY).

4 A * STEM LENGTH SIZE 39.
B STEM LENGTH OF OTHER SIZE.

SPECIFY NEW OPTION REQUIRED,
"RETURN" IF ALL ARE OK.

Figure 3.2(b) Display showing the options available, TRAD notation selected.

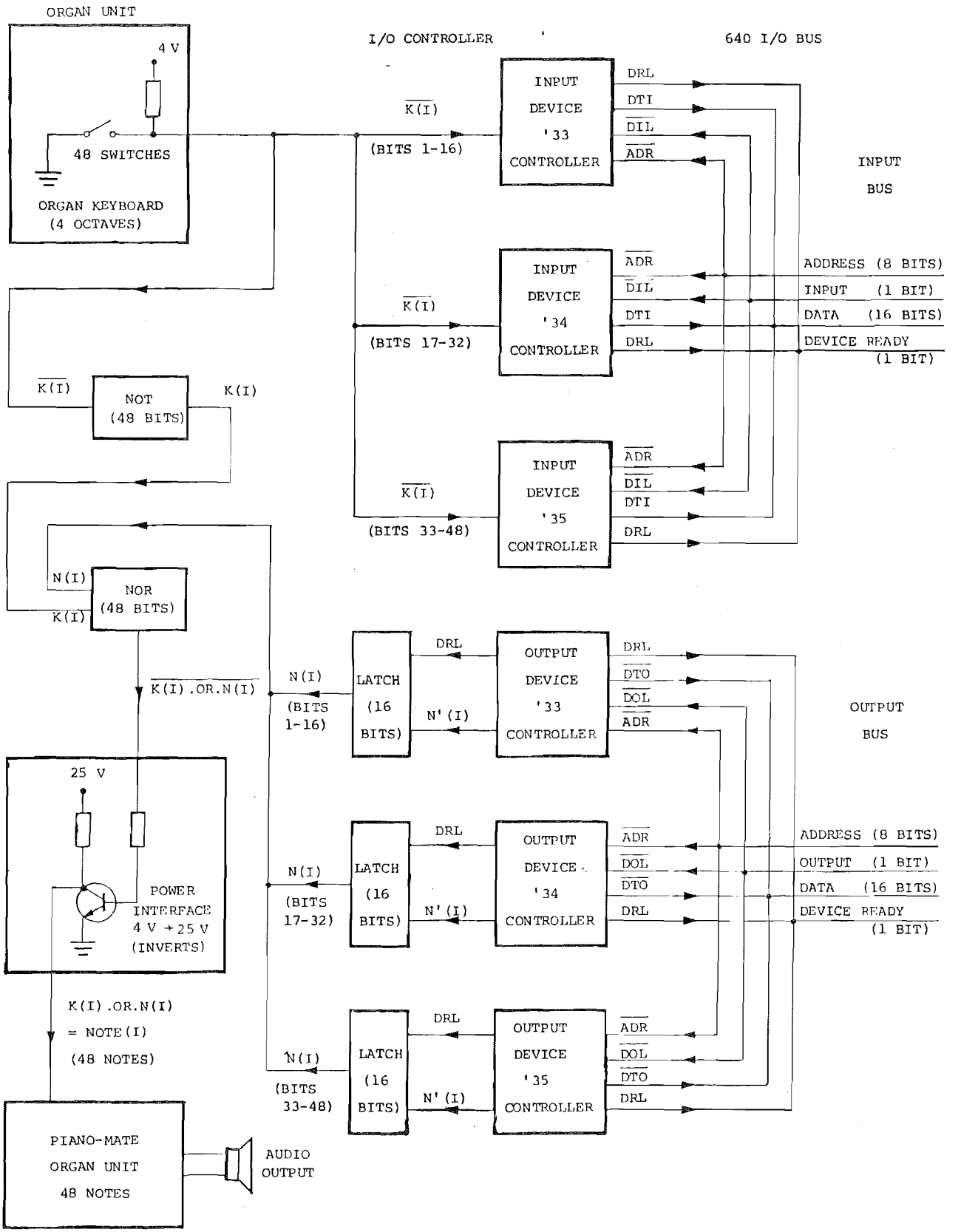


Figure 3.3 RTL I/O Interface and organ - schematic. See also Table 3.2.

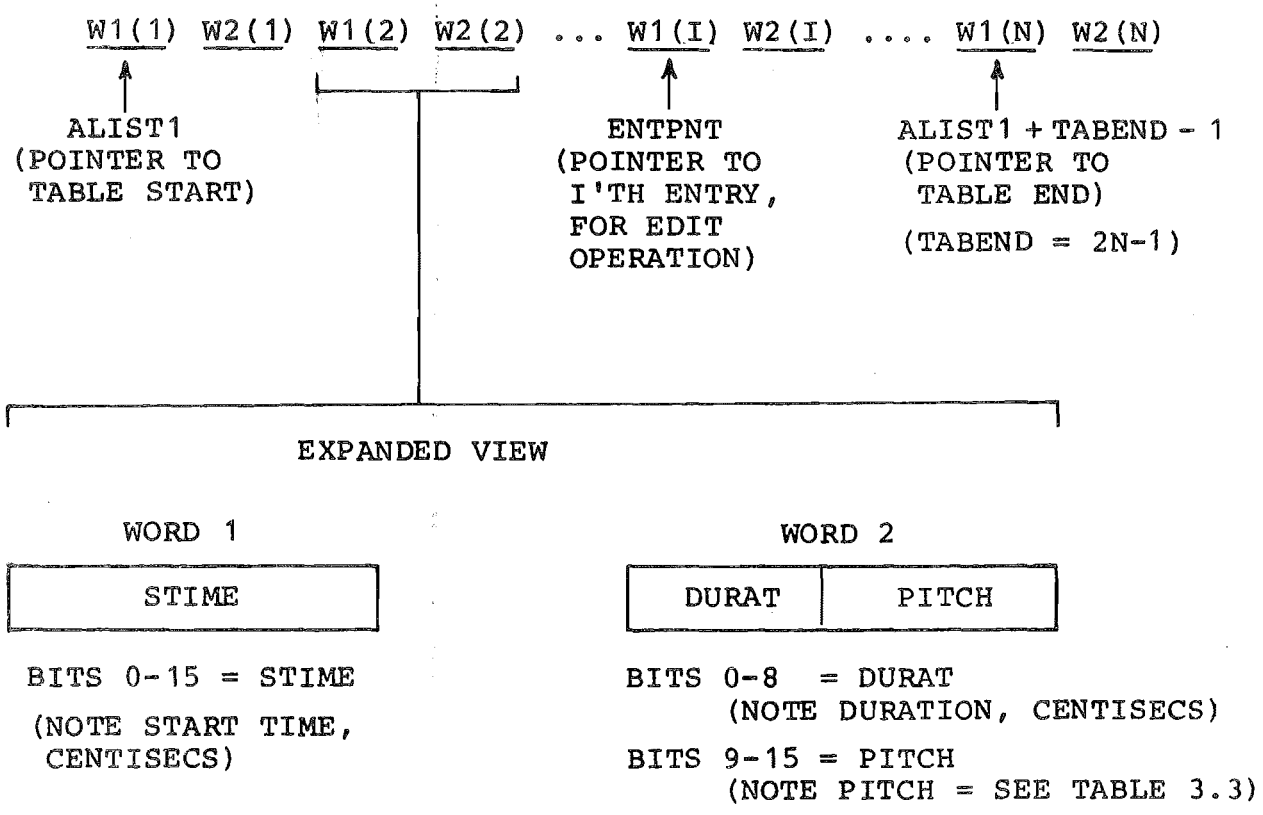


Figure 3.4 MOD data structure, showing the sequential table form and illustrating parameter packing.

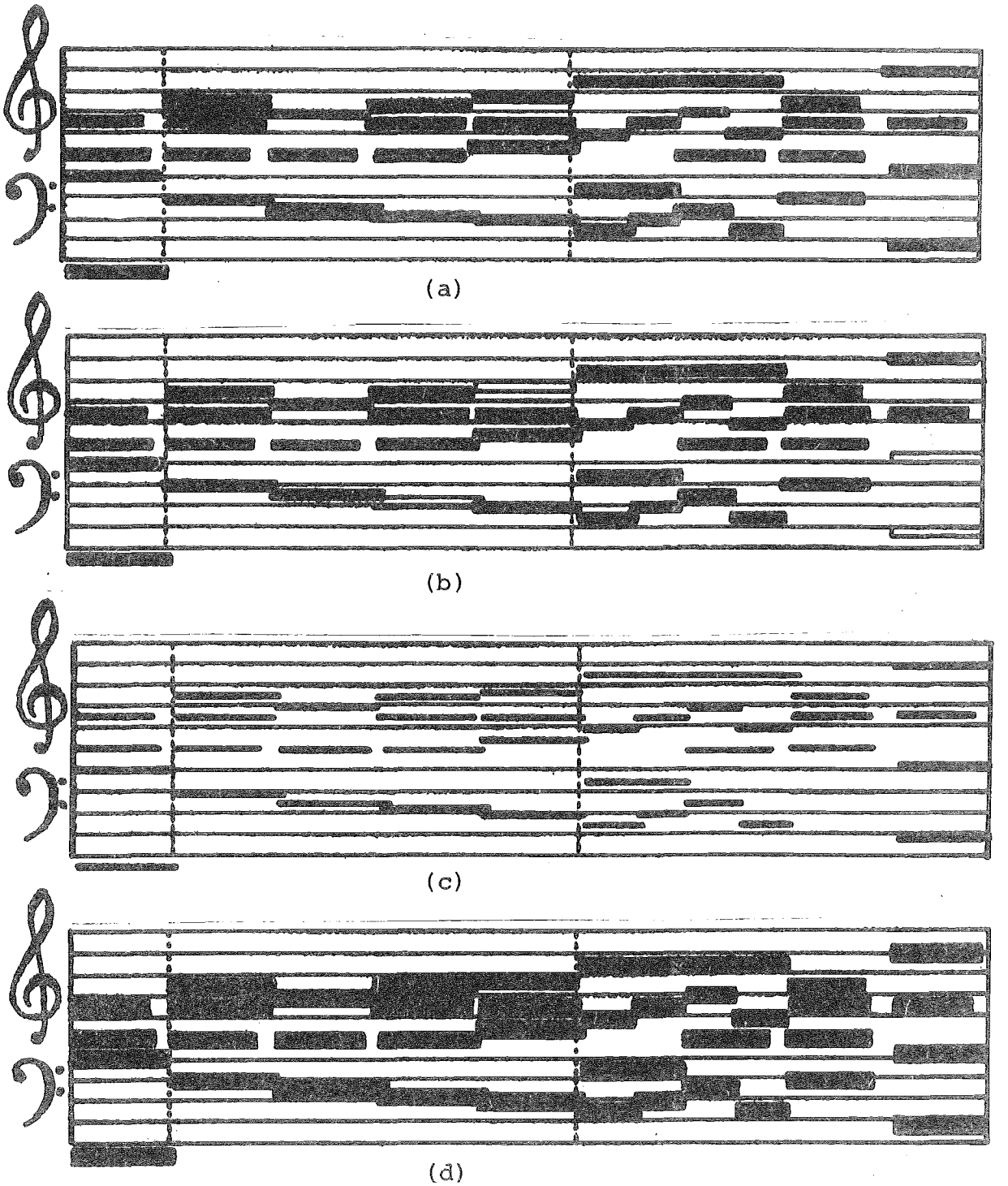


Figure 3.5 Example of MOD display.

(a) Display option 3A. (Black and white keyboard notes not distinguished). Standard note "block" width.

(b) Display option 3B. (Black and white keyboard notes distinguished). Standard note "block" width.

(c) Display option 3A, narrow note "blocks".

(d) Display option 3A, wide note "blocks".

CHAPTER 4

CONVENTIONAL MUSIC NOTATION - TRANSCRIPTION, DISPLAY AND EDITING

4.1 INTRODUCTION

The MOD music notation which is described in Chapter 3 is ideally suited to those applications which require nuances of performance or tempo to be depicted. Such applications include teaching keyboard technique (Tucker, Bates, Frykberg *et al.*, 1977), displaying and manipulating music during the composition process (Frykberg and Bates, 1977), and editing music which is to be performed under computer control (e.g. by electronic synthesiser). However, despite its advantages of simplicity and isomorphism with the corresponding aural music, MOD is not suitable as a performance notation for use by musicians. As Read (1974) observes while commenting on similar notations, reading such music requires complete re-education of the eye.

The conventional music notation has evolved over centuries into a highly efficient symbolic representation of sound sequences (Gamble, 1923; Ross, 1970; Read, 1974). Consequently it is oriented specifically towards the performance of music by human performers, and most musicians (at least those of Western civilisations) are trained in its use. Thus, the range of applications of any computer-aided

music facility can be significantly enhanced if TRAD display is incorporated. Moreover, for such a system to be widely used and accepted by musicians, it must be able to produce written music in a notation which is acceptable to them.

Historically, the initial aim of this project was to develop a fast music transcription system which produces TRAD music typescript of a standard suitable for subsequent printing by photo-lithography (Tucker, 1972). As mentioned in Section 3.1, the main difficulty encountered by the 1972 undergraduate project was the development of a reliable music input system using the piano/organ keyboard. Once this technical problem was overcome the inherent difficulties associated with transcribing the recorded information into a form suitable for TRAD display became apparent. These problems arise from several sources. Firstly, it is difficult for even a skilled pianist to play accurately and consistently, so that note durational values can be recognised directly. This is called the "roundoff" problem, because its solution requires a "rounding off" procedure which reduces or eliminates the temporal variations between the actual and nominal start times and duration values of different notes or chords. These nominal values are called the "ideal" values, since they correspond to a perfectly literal performance during the record process. Secondly, TRAD notation is not isomorphic to an "ideal" performance. Numerous notational conventions exist which aid the human performer but are not reflected in the corresponding aural music. Examples of such notation conventions are ties and ledger lines, the directions of

note stems and tails, the grouping of notes by beams, and the assignment of accidentals which distinguish enharmonic equivalents.

The initial approach to the "roundoff" problem was to use a variable speed keyboard sampling rate during the record procedure. By using a special keyboard sampling clock which is synchronised to a metronome, it was hoped to be able to record note durations directly in a binary code. Thus the sampling clock was calibrated so that a semibreve (whole note) corresponds to $128 = 2^7$ samples, a minim (half note) corresponds to $64 = 2^6$ samples, etc. This direct relationship between the keyboard sampling rate and the "ideal" TRAD duration values has several advantages. Firstly the binary duration code relates elegantly to the TRAD duration values - each note in any combination of dotted and tied notes is represented by a "1" bit in the position corresponding to its duration. For example, a double-dotted crotchet is represented as $2^5 + 2^4 + 2^3$, or binary 111000. The only exception to this rule occurs when compound rhythms or triplets are encountered. Secondly, because of the simple coding structure, it is easy to "round" actual durations using standard numerical truncation and roundoff procedures. However, several limitations of this approach soon become apparent. The most critical of these is that note attack (i.e. start) times seem to be much more important than note durations, and are consequently played more precisely. Thus the actual durations of notes which are nominally the same exhibit considerably greater variations than the corresponding intervals between the

start times of those notes. This observation is even more significant when such deliberate variations in style as legato and staccato are considered. Because of this, the comparatively simple duration roundoff procedure outlined above does not work satisfactorily. A more suitable approach is to round note durations so that each note end is aligned to the nearest note start time, with suitable allowance for the possibility that a rest may be required. This approach is used in the current roundoff procedure, which is described in Section 4.4.

A less severe limitation of the initial duration coding scheme is that the metronome must be used - this is irksome to many musicians. In addition, the stored duration values are quantised to levels which are comparatively coarse (cf. the 100 Hz fixed sampling rate finally adopted - see Chapter 3). This means that nuances of performance are not faithfully recorded. Since it is desirable that the TRAD and MOD portions of the system be fully compatible, the variable speed sampling rate scheme was abandoned.

A major difficulty encountered with the development of the TRAD display (using the ideal, rounded note table as input data) was the organisation of the software and the design of a compact, non-redundant intermediate data structure. This difficulty arises from the fact that the TRAD notation is highly redundant (cf. Böker-Heil, 1972) and that many of the notation conventions are dependent upon the musical context (Read, 1974). Two approaches to the display problem are possible. One approach is to generate all the required display parameters in successive passes, and thus

construct a complicated and highly redundant display data table which is subsequently used as input to a procedure which draws the appropriate symbols. This method is called "display from table". The other approach is to generate the display parameters for each note (or chord, or beamed note cluster) and to draw the appropriate symbols as soon as the required parameters are generated. This method is called "display by rule". The limited availability of core storage which has already been mentioned (Section 3.2) requires that the "display by rule" approach be used as much as possible. However, it is very unwieldy to incorporate into "display by rule" the level of contextual information which is required for a satisfactory display. For example, musical punctuation requires that the positioning and separation of adjacent notes take into account the presence or absence of such symbols as accidentals or dots, and the duration values of the preceding notes. The initial display system which was developed during 1972 and 1973 ignored such contextual requirements. It was designed to process and draw each note or rest independently of the surrounding symbols, and consequently produced quite unsatisfactory results. The present system uses a combination of display "from table" and "by rule" to effect a compromise between the level of redundancy and contextual information contained in the display data table, and its compactness.

A further requirement of the display system is that both "typescript editing" and "performance editing" be accommodated. The latter is comparatively easy to implement, whether display "from table" or "by rule" is

used, because the input note table data (such as pitch and duration) is modified. However, typescript editing (which alters the presentation of symbols, e.g. stem direction or length) cannot be directly incorporated if the display is "by rule". The method adopted in the present system is to generate an "exceptions table" (called the secondary note table) for all edited parameters which cannot be altered in the input data table. The secondary note table entries over-ride the display parameters which are generated "by rule".

The remaining sections of this chapter describe the roundoff, transcription, display and edit portions of the TRAD display system. The various stages of the display and editing procedures are illustrated by examples, which also demonstrate the practical application of the system in generating music typescript for subsequent printing.

A brief review and bibliography of the historical development of music printing techniques is given by Howarth (1977a). Kassler (1977) gives a more detailed discussion of the commercial advantages of computer-assisted music printing over the traditional labour-intensive methods, which include engraving, music typing and autography (i.e. reproduction from hand-copied music). It is sufficient here to observe that the advantages of applying computer-assisted music typesetting to both the efficient storage and reproduction of written music are now well established. Kassler's (1977) conclusion that an organ keyboard is not a suitable input medium for subsequent display is an indication of the complexity of the transcription problem.

Nevertheless, it is asserted here that the system described in this chapter overcomes many of the deficiencies of previously reported typescript generation systems. In addition, it is sufficiently flexible in its design that further improvements can be made without requiring drastic reorganisation.

4.2 TRAD TYPESCRIPT PRODUCTION - AN OVERVIEW

The TRAD transcription, display and editing system is a subsystem of the "Piano Typewriter", which is described in Chapter 3. The TRAD system software is organised in three core-image phases - TRADT1, TRADT2 and TRADT3 - which are interfaced to the Piano Typewriter system in the manner described in Section 3.9. The organisation of the component modules within these phases is summarised in Table 3.5.

The procedures required to generate TRAD music typescript in the form of a master copy suitable for reproduction (by photo-lithography for example) are conveniently divided into distinct processes. These are: recording, bar-line insertion, transcription, display, editing, and final hard copy generation. First, the music is played carefully on the organ and recorded in the MOD data structure (see Sections 3.4 and 3.5). The organ may be played at any speed - preferably slowly - and it is important that the tempo be consistent within each bar (measure). However, substantial tempo variations are permitted between different bars. Thus, the interval which corresponds to (say) a semiquaver in one bar can be the same as the interval which corresponds to a crotchet (say) in

another bar. This flexibility is very useful when a complicated or difficult passage of music is being recorded. Bar lines are added to indicate the time instants which correspond to the transition between adjacent bars. This may be done using the foot switch during record, by using the foot switch in conjunction with playback (the "B" command - see Table 3.1), or using the joystick and MOD display (the "I,BL" command). Bar lines should be inserted carefully because the time instants to which they correspond are critical in the roundoff procedure. The user may display or play back the recorded music, and carry out performance editing using the MOD display and editing facilities. Alternatively, such performance editing can be deferred to the TRAD editing stage.

The next stages of TRAD typescript production are transcription and display - the distinction between them is transparent to the user. These are initiated by selecting the TRAD display option (using the "X" command), and typing "S" (or "SCORE"). This causes the phase TRADT2 to be loaded into core, and control to be transferred from the PIANO master controller to the TRAD controller. The latter incorporates an interpreter, and controls the entire transcription, display and editing processes.

The recorded note table in the MOD data format is first transcribed into a TRAD primary note table. The format and content of the latter are described in Section 4.3, and the transcription process is discussed in Section 4.4. The TRAD primary table is an efficient coding of the required TRAD score, and contains no information about such

display details as stem directions or lengths, or the existence of ledger lines. The data transcription is performed in several passes. The MOD note table is first rounded to the corresponding "ideal" performance. The resulting rounded MOD note table is then translated into TRAD primary note table form. Subsequent passes elaborate this initial TRAD primary note table, by inserting information about accidentals and rests.

When the TRAD primary note table has been generated, the display process starts. This is performed in three distinct stages for each page - page initialisation, justification and symbol display - which are described in detail in Section 4.5. The page initialisation module erases the display screen, draws the stave lines and treble and bass clef signs, and numbers the page. The justification stage allocates the X co-ordinates for all vertical clusters on the current stave group, taking into account the normal conventions of musical punctuation. These co-ordinates are stored in a temporary table and are scaled so that the final bar line coincides with the right-hand end of the stave group. The term "vertical cluster" here means a cluster of vertically-aligned symbols - e.g. a note (including its associated accidental symbol and dot(s)), a chord, or a rest. The term "stave group" is herein used synonomously with the correct musical term "stave system" to denote a group of staves which extend from the left-hand to the right-hand margin of a page and for which corresponding bar lines are aligned vertically. For example, there are two staves to a group for piano music and four for a

string quartet score.

The symbol display procedure uses the TRAD primary note table and the table of scaled X co-ordinates to generate the detailed display parameters. Each note or rest entry in the primary note table is checked for an edit entry in the secondary note table, and if this exists the display parameters are read. If no edit entry exists the display parameters are generated "by rule", taking into account the different requirements imposed by the display format used. Thus, the four part harmony and the single instrument display options (see Figure 3.2(b)) use different algorithms for generating stem directions and assigning vertical co-ordinates from pitch values. As soon as the necessary parameters are generated the component primitive symbols are drawn. For example, a dotted quaver at C#4 (i.e. C# above middle C) is a composite symbol composed of a black (i.e. solid) note head, a sharp symbol, a dot, a stem, a tail and a ledger line. When all the symbols on the current stave group are drawn the temporary storage tables are re-initialised and the next stave group is justified and displayed. When the current page is completed the TRAD controller types "$\leftarrow$" and accepts and interprets subsequent commands.

TRAD editing is performed using operator procedures similar to those required for MOD editing using the display and joystick. These procedures cause the appropriate entries in the primary note table to be altered, or a new entry to be established in the secondary note table (unless a secondary note table entry already exists, in which case

this entry is suitably modified). A further table which specifies the number of bars required on each stave group permits manual control over the layout of each page. This "pagination table" is initialised to a default value of three bars per stave group, and may be altered using the "J,n,m" command (see Table 3.1). TRAD editing procedures are discussed in Section 4.6.

The display may be annotated using the "I,T" command (see Table 3.1). This annotation facility is also available with the MOD display, and permits messages comprised of alphanumeric characters to be written at specified positions on the page. The character size is controllable (in discrete steps), and a message deletion facility is provided using the "D,T" command.

When the necessary editing has produced the desired final display, a hard copy is required. This can be obtained using the Tektronix hard copy unit which is incorporated in the graphics system of our EAI 640 computer. Unfortunately this system suffers from a number of hardware limitations which result in a comparatively low quality music typescript. Howarth (1977a, 1977b) has developed a music plotting system called GUTENBURG which overcomes these hardware limitations by using the CALCOMP plotter in the Computer Centre of the University of Canterbury. GUTENBURG is essentially a FORTRAN IV transcription of the TRAD display software, and is intended as a high quality plotting system which operates in "batch" rather than "interactive" mode. To use this music plotting facility, the user types "$W" (write tape), which causes the TRAD system to punch a paper tape.

This paper tape contains the various TRAD data tables as well as such system parameters as the currently-selected options. The paper tape is then taken to the Computer Centre, together with a set of punched cards which control the operation of GUTENBURG. The resulting plotted music typescript is usually available for collection a few hours later. This facility is discussed further in Section 4.7.

The organisation and functioning of the modules which perform these various stages of music typescript generation are now described in detail.

4.3 TRAD DATA STRUCTURE

In this section are described the structure and format of the TRAD primary and secondary note tables, the pagination table which specifies the display layout, and the text table in which is stored the display annotation. The core storage and COMMON conventions used for these tables are summarised in Table 3.4. The interrelationship between them is depicted in Figure 4.2. In addition is described the format of the temporary beam table. This is constructed to store the display parameters of each note in a beamed cluster until all notes in the cluster have been processed and the cluster can be drawn.

4.3.1 Primary Note Table

The primary note table is a sequential rather than linked list structure, to economise on storage, processing time and manipulative complexity (cf. Fredlund and Sampson, 1973; also see Section 3.4). The syntax of the primary note

table is defined in Backus Naur form in Table 4.1. Each entry in the table occupies two (16 bit) words. These two words are each declared as integers, so that the table is essentially a two dimensional array. The first and last entries must each be a <bar line> . A <basic note> is represented by its <pitch> and <duration> , which are integers whose code values are given in Tables 3.3 and 4.2 respectively. If an accidental is associated with the note, the <basic note> entry is preceded by the appropriate <accidental> entry. If the note is a member of a beamed cluster, or is tied to another note or split across a bar line, then this is indicated by the corresponding <note prefix> entry. Observe that this <note prefix> entry incorporates a <cluster identifier> which is the same for all components of a particular (beamed or tied) note cluster. The manner in which other components of a score are encoded is evident from Table 4.1. To avoid confusion, the codes which correspond to the terminals of the BNF description are listed separately in Table 4.1(b).

Two aspects of the primary note table coding need further explanation. Firstly, the <shift operator> is used to distinguish between adjacent <vertical clusters> , so that chordal groups of notes and rests can be easily distinguished. This is especially useful to the algorithm which allocates X co-ordinates during the justification procedure. Secondly, each <shift operator> and <bar line> entry has an associated <duration> entry. This <duration> corresponds to the time interval from the previous <shift operator> or <bar line> entry (whichever

is nearest), and is coded according to Table 4.2. Thus the start time of any note may be calculated by summing the <duration> values of all <shift operator> and <bar line> entries which precede the note entry. This information is redundant for a fully coded score, but is required at intermediate stages of the primary note table generation. For example, the duration values which are assigned to rests require knowledge of the start times as well as durations of the adjacent notes. In addition, several algorithms invoked during the justification and display procedures require knowledge of the shortest duration value present in a <vertical cluster> . This information is obtained by examining the <duration> value of the following <shift operator> or <bar line> entry, so that a sub-scan is not required.

A convention which is not indicated in Table 4.1 is that all notes and rests within each vertical cluster are arranged in pitch descending order. This convention is useful when notes or rests must be allocated to a particular voice or part - for example when four part harmony music is being displayed (see Section 4.5.3).

The encoding of a score in the primary note table is illustrated in Figure 4.1.

4.3.2 Secondary Note Table

In contrast to the sequential structure of the primary note table, the secondary note table is implemented as an inverse linked list. The reasons for this are twofold. Firstly, the secondary note table is usually sparse, in the sense that many of the notes and rests in the primary note

table do not have a corresponding entry in the secondary table. Secondly, typescript editing is in general a "random" rather than "sequential" process, so that entries generated chronologically in the secondary table do not follow the same sequential ordering which is used in the primary note table.

Each entry in the secondary note table occupies three (16 bit) words. The first word contains a pointer which specifies the corresponding primary note table entry. The second and third words contain six display parameters, which are packed using the convention defined in Figure 4.3. A suite of fast 640 ASSEMBLY language modules is available to manipulate the secondary table, and to pack and unpack the display parameters. In this way the typescript editing procedures are simplified, and the inverse linked list nature of the secondary note table is virtually transparent.

4.3.3 Pagination Table

As mentioned in Section 4.2, a useful interactive editing feature is the "J,n,m" command (see Table 3.1) which permits manual control over the layout and pagination of each displayed page. This control is achieved by using a pagination table which specifies the number of bars to be allocated to each stave group. The pagination table is organised as a two dimensional table, which is indexed by page and line number.

4.3.4 Text Table

Annotation of the display with text is achieved by storing in a "text table" the required text, organised in

discrete "messages". The term "message" here refers to a single line of contiguous alphanumeric characters. Standard 8 bit ASCII alphanumeric code is used and the text is packed two characters per (16 bit) word, observing the A2 format convention. The end of each message is defined by the RETURN character.

To provide convenient access to each message, a separate message "directory" is used. This directory is called the "text pointer table", and consists of a five word entry for each message. These five words define the following message parameters: the page number on which the message is required, the size of all characters in the message, the X and Y co-ordinates of the first character, and the pointer which defines the location of the first character in the text table.

The present system provides storage for only 10 messages, or 200 characters (whichever is reached first). This is insufficient for many applications, but is a limitation which is easily extended. A more serious disadvantage of the facility is that the stored message co-ordinates are absolute rather than relative. Thus, it is often convenient to align a message to a particular symbol (e.g. a note or bar line) so that subsequent editing or pagination changes do not alter the relativity between the symbol and message. This applies particularly to dynamic indications and song words.

The present music plotting system GUTENBURG does not handle text, so that all annotation of the final music typescript must be added manually.

4.3.5 Temporary Beam Table

The temporary beam table is constructed during the actual display process (see Section 4.5.3). Its function is to store the display parameters of each note in a beamed cluster until all notes in the cluster have been processed. This temporary storage is required because the final allocation of some display parameters (such as stem direction) cannot be made until information is available about all the notes in the cluster. When the entire cluster has been processed and drawn, the corresponding entries in the temporary table are erased and the storage space made available for another beamed cluster.

The table is implemented as a three dimensional array of integers called LINKD (I,J,K). Storage is provided for up to 5 beamed clusters (indexed via I), each consisting of up to 9 notes (indexed via J = 2 to 10. In addition, parameters relevant to the entire cluster are stored in the entry which is indexed at J = 1. The cluster parameters (indexed via K) are: the <cluster identifier> entry from the primary note table, the J value of the most recently entered note, and a flag which indicates the stem direction for all notes in the cluster. This latter is not normally assigned until all notes have been processed, but may be set up by a secondary note table entry. The <cluster identifier> entry is used to assign the correct index I to each new note entry. The parameters stored in each note entry are indexed via K and consist of: pitch and duration packed into one word, the pointer which specifies the address of the corresponding note entry in the

primary note table, the X co-ordinate at which the note symbol is to be drawn and a flag which specifies the type of accidental required (if any). The primary note table address is required so that individual notes within a beamed cluster can be located by the editing procedures.

4.4 TRANSCRIPTION - CONVERSION OF THE DATA BASE

The transcription process which generates the TRAD primary note table from the MOD note table operates in several stages - roundoff, translation and elaboration. These procedures are now described in detail.

4.4.1 Roundoff

The roundoff procedure "idealises" the start time and duration value of all notes in the MOD note table, and retains the MOD data structure (see Section 3.4). This process operates bar by bar, and requires knowledge of both the time signature and the instants at which each bar begins and ends. The latter are defined by the "bar lines" which are inserted using one of the three methods described in Section 4.2. The time signature is specified by the operator in response to the request "ENTER TIME SIG, "N/M"←". It is stored as the two variables NBEATS (which defines the number of beats per bar) and BEATYP (which specifies the duration value of the beat note).

In addition to this information are required two threshold parameters, which govern the extent to which start times and durations are rounded. These two parameters are the "roundoff severity factor" and the "minimum rest length".

The former is specified as an integer, in response to the request "ENTER ROUNDOFF SEVERITY FACTOR (1 TO 20). 1 IS FINEST, 20 IS COARSEST". This is converted to the range 0.1 to 2.0, and stored as the floating point variable RNDFAC. The minimum rest length is similarly specified as an integer in the range 1 to 16 (which correspond respectively to duration values of one semiquaver and one semibreve - see Table 4.2). This value is stored as MINRST. The operation of these two parameters is described below. Since they are specified by the user, rather than predetermined within the system, the user is given control over the extent to which notes are rounded. This also permits the transcription system to accommodate variations in the style and accuracy of the original playing.

Each bar is rounded using a two pass procedure. In pass 1 the bar is divided into a sequence of "time slots". The duration of each time slot corresponds to a semiquaver (1/16th note). This interval has been chosen as the quantisation interval (denoted by QTIME), since it is the smallest interval which can be resolved under normal playing conditions. QTIME is measured in centiseconds and is computed from the formula:

$$QTIME = BEATVL * BEATYP / 16 \qquad (4.1)$$

where

$$BEATVL = BRLNTH / NBEATS \qquad (4.2)$$

(* and / here denote the operations of multiplication and division, respectively). BEATVL is the interval (in centi-seconds) corresponding to an ideal beat duration, and BRLNTH

is the interval between the "bar line" instants which define the beginning and end of the current bar. Floating-point rather than fixed-point arithmetic is used, to reduce numerical truncation errors.

The start time of each note in the current bar is next adjusted to align with a "time slot" boundary. This is performed by associating with each time slot a threshold window, whose width is computed in the manner described below. All notes whose start times lie within the threshold window are aligned to the corresponding time slot boundary. The initial threshold window width is given by:

$$2 * \text{THRESH} = 2 * \text{QTIME} * \text{RNDFAC} \quad . \qquad (4.3)$$

All time slot boundaries which correspond to beat (and bar) boundaries are examined first. The threshold window width is then decreased to 2/3 of its current value and the time slot boundaries midway between those already processed are examined next. This procedure continues until all time slots have been processed. To prevent unnecessarily fine resolution and excessive computation, the threshold window width is not permitted to become smaller than QTIME.

When all note start times in the current bar have been aligned, pass 2 commences. The duration of each note is adjusted by requiring that the note ends on the nearest interval boundary on which another note starts, unless the difference between the "actual" and "ideal" end times exceeds the interval which corresponds to MINRST (i.e. MINRST * QTIME). When the latter condition occurs, the note duration is rounded to the nearest integer multiple of QTIME.

In addition, any note which extends by more than QTIME/2 past the end of the current bar is "split" into two notes, such that the bar end instant defines the end of the first note and the beginning of the second. This second note segment is processed with the next bar. All split notes are identified by a "split note tag" prefix entry.

The current rounded bar is stored in a temporary buffer.

At present, no provision is made for triplets - if these are present they are rounded to the nearest multiple of a semiquaver. However, the display edit procedures (e.g. change duration and insert beam) can be used to correct this deficiency, since triplet durations are represented using the duration codes listed in Table 4.2.

4.4.2 Translation

The current rounded bar in MOD note table form is next translated into the equivalent TRAD primary note table, in which time intervals are represented using the parametric TRAD duration code (see Table 4.2) rather than their real time values in centiseconds. This translation procedure is straightforward. Start times are converted from absolute to incremental form, to obey the conventions described in Section 4.3.1. The conversion from real time to duration code is performed by dividing real time by QTIME (since the duration code given in Table 4.2 corresponds to integral multiples of a semiquaver).

When the current bar has been translated, the next bar is rounded and translated. In this way the initial TRAD primary note table is constructed bar by bar.

4.4.3 Elaboration

When the roundoff and translation process is completed, the initial TRAD primary note table is elaborated by inserting accidentals and rests, and grouping notes "horizontally" with beams. The procedure which allocates accidentals was designed by Lamb (1977), and is an extension of Longuet-Higgens and Steedman's (1971) key signature analysis algorithm. This procedure determines the overall key signature of the passage, and any modulations which occur. The accidental(s) appropriate to the modulation is assigned, and the required <accidental> entry is inserted by invoking the TRAD edit "insert entry" procedure.

Horizontal grouping of notes so that ties, beams and rests can be assigned is straightforward for a single musical line, but is difficult when the music has a vertical as well as a horizontal structure. The main problem is how to track temporal and pitch transitions in the absence of explicit "voice crossing" information. This can be achieved at a rudimentary level by neglecting the possibility of voice crossings. Thus each voice is identified at any horizontal position by its vertical position, taking into account notes from previous chords whose durations extend to this horizontal position. This technique of voice identification is also used for the automatic assignment of stem directions and note pitch origins (i.e. whether pitch is assigned relative to the bass or treble stave - see Section 4.5). Algorithms which assign beams, ties and rests using this approach are currently being designed by Susan Frykberg. At present however, this stage of the

elaboration process has not been implemented, and must be performed manually using the editing procedures described in Section 4.6.

4.5 TRAD DISPLAY

Three distinct stages - page initialisation, justification, and symbol drawing - are required for the generation of each displayed page. The organisation and operation of the component modules which comprise these stages are now described in detail.

4.5.1 Display Organisation

The display task is organised in two core image phases in the manner outlined in Table 3.5 and Figure 4.4. A single control program (subroutine SCORED) which is common to both phases supervises the overall display of each complete page and the transfer of executive control between phases. The first task performed by SCORED is page initialisation. This is primarily a housekeeping operation which sets up various temporary storage tables, pointers and control variables. It also erases the display screen, draws the stave group lines and treble and bass clef signs, and numbers the page in preparation for the actual music display. This display page preparation task is performed by subroutine STVDW2, which uses the vector drawing software described in Section 4.5.4.

The display task proper then commences. The number of bars required in the current stave group is read from the pagination table using subroutine JSTSET, and the

allocation and justification of symbol X co-ordinates is performed in the manner described in Section 4.5.2. The display parameters are then generated for each note or rest in the current stave group, and the appropriate symbols are drawn. This procedure is discussed in Sections 4.5.3 and 4.5.4. When all symbols on the current stave group have been drawn the next stave group is similarly processed, until all stave groups on the current page are displayed. Finally, the text table is examined and any annotative messages required for this page are written, using the procedure described in Section 4.5.5. Control is then transferred to the TRAD edit controller which accepts and interprets the next operator command (see Section 4.6).

4.5.2 Justification

Subroutine LNJUST controls the justification procedure. The primary note table is scanned and an X co-ordinate is allocated to each <shift operator> and <bar line> entry. The co-ordinate which is allocated depends upon the smallest duration of the notes or rests in the preceding vertical cluster, and is adjusted to compensate for any accidentals present in the current vertical cluster. In this manner the rules of musical punctuation are observed (Ross, 1970; Read, 1974). The allocated X co-ordinates are stored in a temporary table. When the number of bars required in this stave group have all been processed, the X co-ordinates are scaled so that the final bar line coincides with the right-hand end of the stave group. This scaling is performed by subroutine JSTSCL.

4.5.3 Generation of Display Parameters

The generation of the detailed display parameters and the drawing of the musical symbols are controlled by subroutine LNDRAW. The justified X co-ordinate for each vertical cluster is read, and the entries which comprise this cluster are successively decoded from the primary note table. If any syntactic errors in this table are detected then an error message is typed and an appropriate error recovery sequence is invoked.

The display parameter values depend significantly upon the type of music being drawn. For example, the rules which govern stem directions and pitch (i.e. whether a note is drawn relative to the treble or bass stave in for example piano music) are different for four part harmony music than for instrumental or piano music. To accommodate these various types of music, a display format option is provided (see Figure 3.2(b)). This option is used as a steering variable which specifies the particular display algorithms to be used. For simplicity in the following description the "single part" option is assumed to be selected - this option is applicable to instrumental music of the kind illustrated in Figure 4.7. If more than one note or rest exists in any vertical cluster then those notes (or rests) are treated independently.

The action taken when each entry in the primary note table is decoded is as follows:-

<bar line> : Draw a bar line at the current X co-ordinate. Start a new vertical cluster as for <shift operator> .

<shift operator> : Start a new vertical cluster - read the next X co-ordinate, and read and decode the next primary note table entry.

<beam start> : Read the next entry. If it is an <accidental> , determine the type (sharp, flat etc.). Read the <note> entry. Test for an entry in the secondary note table corresponding to this note - if an entry exists read: stem length, stem direction, note pitch origin, four part harmony flag, horizontal shift for note, and horizontal shift for accidental. If no secondary note table entry exists, set the above listed parameters to their default values for calculation later. Set up a new entry in the temporary beam note table using subroutine SETLNK - store the cluster identifier, accidental type, pitch, duration, X co-ordinate, note pitch origin (if not default value), stem direction (if not default value), and primary note table pointer (this permits the location of the note by the edit procedures). Read and process the next primary note table entry.

<beam middle> : Process as for <beam start> , but associate the note display parameters in the temporary beam table with those of other notes in the same beamed cluster.

<beam end> : Write the entry in the temporary beam table as for <beam middle> . Then generate the display parameters and draw the component primitive symbols using subroutine NLINK. There are two cases:

Case (i) Stem direction is the default value. Calculate the stem direction and pitch origin using subroutine STMDEC - the operation of this is described in the <note> entry

processing.

Case (ii) Stem direction is not the default value. Use this value of stem direction and the current pitch origin (these two parameters are set up by a secondary note table entry associated with the last note of the beamed cluster).

If the stem length is the default value, compute the effective stem length for the first and last beamed notes so that all note heads are on the same side of the beam, and the note head nearest to the beam is no closer than a preset threshold. Otherwise, use the specified stem length. Draw all component primitive symbols as for <note> , but adjust each stem length so that the stem ends on the beam. When all notes in the beamed cluster have been drawn, draw the beam.

<split start> :
<split end> :
<tie start> :
<tie middle> :
<tie end> :

These entries are ignored in the present system, because tie and slur symbols have not yet been incorporated. As above. As above.

<accidental> : Read the accidental type and set up the accidental flag which is used in processing the associated note. Read the next primary note table entry - it must be a <note> . Process this <note> as described below.

<note> : Test whether this note has a secondary note table entry. If so read: stem direction, pitch origin, four part harmony flag, stem length, X shift of note, X shift of accidental. Draw the required symbols as described below. Otherwise, calculate these parameters by rule. Set X shift of note and accidental to zero, set stem

length to the standard value, and calculate the pitch origin and stem direction using subroutine STMDEC. STMDEC assigns the pitch origin to the treble stave set if the pitch is C4 or above, otherwise the origin is assigned to the bass stave set. The stem direction is set to down if the pitch is between D3 and B3, or above A#4. Otherwise the stem direction is set to up. STMDEC will also handle clusters of notes (e.g. beamed clusters) by calculating the mean pitch of the cluster, and applying to this the same decision criteria. The Y co-ordinate of the note head centre is calculated using subroutine YPITCH. A table look-up procedure is used, and the resulting Y co-ordinate is adjusted in accordance with the pitch origin flag. The effective X co-ordinate is established by summing the vertical cluster X co-ordinate and the note X shift value. The note head symbol, plus accidental and dot symbol(s) (if any) are then drawn using subroutine DRAWHD. The type of note head - i.e. black (solid) or white (hollow) and the dot(s) required are determined from the note duration. If the note head sits on a stave line and a dot is required, the Y co-ordinate of the dot symbol is adjusted upward (to clear the stave line) using subroutine YMODOT. Any ledger lines which are required are drawn next, using subroutine DWLDGR. Finally, the stem and tail(s) are drawn using subroutine DRWSTM, which uses the note duration to decide whether a stem and/or tails are required. The various component symbols are drawn using the modules described in Section 4.5.4.

<rest> : The rest symbol and any dot(s) required are

drawn using subroutine DWREST. The X and Y co-ordinates are calculated as for <note> .

When the four part harmony option is set, the same procedures described above are used, but an additional stage is included to determine which voice the current note (or rest) should be assigned to. This module (subroutine FSATB, which is embedded within LNDRAW) sets the four part harmony flag to indicate whether the voice is S (soprano), A (alto), T (tenor) or B (bass). The four part harmony flag then dictates both the pitch origin and the stem direction - thus S and A notes are drawn relative to the treble stave with stems up and down respectively, while T and B notes are drawn relative to the bass stave with stems up and down respectively.

The operation of FSATB is as follows. When the processing of a new vertical cluster starts, a counter KOUNTP is set to zero. Each time a <note> or <rest> entry is read, KOUNTP is incremented (remember that the pitches of entries within each vertical cluster are ordered with pitch descending). KOUNTP is thus an initial estimate of the voice to which the note (or rest) should be assigned. Now if all notes in the previous vertical clusters have durations which are the same within each vertical cluster, no note from a previous vertical cluster will still be operational. In this case KOUNTP indicates correctly the voice to which the current note (or rest) is assigned, assuming that no "voice crossing" has occurred.

However, this is not generally the case, and a check is required to see whether a note (or rest) corresponding to the current voice estimate is still continuing from a previous vertical cluster. This check is performed by establishing a "running store" of note durations indexed by voice number, so that when a new vertical cluster starts the effective remaining duration of the note (or rest) for each voice may be read directly. This "running store" is interrogated to determine whether the current voice estimate KOUNTP has a continuing note (or rest). If so, KOUNTP is incremented and the new voice estimate is similarly tested. If not, then KOUNTP is taken to be the correct voice and the current <note> or <rest> entry is assigned accordingly. The "running store" is then updated. Should more than four <note> or <rest> entries be detected in a single vertical cluster, then an error message is typed, entries after the fourth are assigned as for voice B (bass), and no "running store" action is taken.

The piano music option has not yet been implemented. Such music is currently displayed using the "single line" display algorithms, which assign independently the parameters of each note or rest in each vertical cluster. Subsequent editing is required to correct the display. Future implementation of this display option is expected to use an approach similar to that of the four part harmony option, but with the assignment of notes and rests to "right-hand" and "left-hand" parts. Initially this part assignment would use C4 as the decision threshold - thus entries with pitch C4 and above would be assigned to the

"right-hand" part. Further sophistication of this criteria is possible (for example the decision threshold could vary adaptively so that the pitch span for each part is not excessive). The vertical grouping of notes and rests within each part is comparatively straightforward.

4.5.4 Symbol Drawing

The use of the high-level symbol drawing routines DRAWHD, DWREST, DWLDGR and DRWSTM is mentioned in Section 4.5.3. These modules decode such information as duration and pitch to decide which component "primitive symbols" are required. They then control the drawing of these primitive symbols at the appropriate positions relative to the composite symbol co-ordinates (which have been calculated elsewhere). In this section is discussed the suite of modules which draw the primitive symbols.

Each primitive symbol is composed of a set of vectors. To facilitate both the manual coding of the vector set and its subsequent drawing, a three tiered symbol display hierarchy was devised. ASSEMBLY language is used throughout for speed. At the lowest level, a suite of eight vector drawing routines [VECT1, [VECT2, ... [VECT8 draws vectors in each of eight directions, which are either parallel to or at a 45 degree angle to the axes of a rectangular co-ordinate system. Each vector drawing routine operates at the display hardware control and I/O level. For speed, the correct display mode is assumed to have already been set - this display initialisation task is performed at a higher level. On entry, the initial X and Y co-ordinates and the vector length are stored in hardware registers.

The vector is drawn as a sequence of discrete dots on the screen, in the direction determined by the particular routine used.

At the intermediate level is a routine called VECTGN, which decodes and draws a "block" of parallel vectors. A "block" in this context is a table which contains an entry for each vector. Each block entry specifies the length of the vector, and the co-ordinates of the vector starting position relative to the start of the previous vector. In addition, the number of vectors coded in the block is specified, and a location [VECTN is initialised to contain the address of the particular vector drawing routine [VECT1 through to [VECT8 which is required. Global variables X and Y are also initialised, to specify the block origin. These initialisation tasks are performed at the highest level. Since all vectors coded in a particular block are parallel, non parallel vectors must be coded in separate blocks. The operation of VECTGN is to decode each block entry, compute the current vector start position co-ordinates and call the designated vector drawing routine.

At the highest level is a suite of routines which contain the (manually coded) block tables required to draw each primitive symbol together with supervisory information such as the vector direction and the origin to be assigned to each block. Each routine also ensures that the correct display hardware mode is established.

The primitive symbols which have been implemented include the treble and bass clef signs, the unstemmed hollow (white) note head i.e. breve or semi-breve, the stemmed

hollow note head (i.e. minim), the solid (black) note head, the dot, the up and down note stem of arbitrary length, the up and down note tail, the various rest symbols (semibreve, minim, crotchet, quaver and semiquaver), the ledger line, and the sharp, flat and natural accidental symbols. Additional symbols can be easily added.

It is worth observing here that because of the vector coding scheme used, symbols of arbitrary size can be created with detail resolution governed solely by the hardware. In this way is avoided Howarth's (1977a) problem of lack of detail in large symbols.

The straight line of arbitrary thickness between two arbitrary points (such as is required for note beams) is drawn using subroutine DTHICK. This module uses the display routine called [JOIN which was developed as part of the EAI 640 display system by Mayson (1971).

4.5.5 Text Drawing

Upper case alphanumeric characters are drawn using the existing display system software, which was developed by Mayson (1971). Each character may be regarded as coded in a 10×12 resolution-element matrix. Characters may also be drawn at sizes which are integer multiples of the basic matrix size, which is designated as size zero. This is also the approach used by Howarth (1977a) to plot the various musical primitive symbols.

The software documentation pertaining to the use of this system is given by Mayson (1971). The author has incorporated this in a high-level module called SCREEN, which writes character strings coded in ASCII, A2 format.

This module is used in conjunction with a text table message decode routine to display stored messages.

4.6 TRAD EDITING

TRAD editing is similar in concept to the MOD editing which is described in detail in Section 3.8. Table 3.1 lists the facilities available and the required operator action.

The edit task is controlled by subroutine TRDCNT, which incorporates an interpreter. Since the PIANO system master control interpreter (see Chapter 3) is not present in the TRAD subsystem phases, TRDCNT also controls other display-related commands such as "S" and "N". When commands relevant to the overall system are encountered (for example file operations) execution reverts to the PIANO system master control interpreter.

Individual notes, rests or accidentals are located using the "invisible display" concept described in Sections 3.7 and 3.8. As with MOD editing, a comprehensive suite of modules is provided to perform individual sub-tasks which are common to many of the edit operations. Thus, individual modules are available to locate a note, rest or accidental symbol which is pointed to by the joystick cursor, to effect an entry change, deletion or insertion in the primary note table, to manipulate the secondary note table so that its inverse linked list structure is transparent, and to insert and delete annotative text messages. Since the structure of the various tables and their operation within the display task

are described in detail in previous sections of this chapter, further description here is not warranted. Full details of the operation of the edit modules are given in the software listings and associated documentation.

The kind of results which are obtained using interactive editing are illustrated in the examples given in Section 4.8.

Experience which has so far been gained with the system suggests that several aspects of the editing procedures should be refined to improve the speed and convenience of the system. Firstly, the present error handling procedures cause the current task to be aborted if an error condition (such as "symbol not located") is detected during an edit task. This approach to error handling is inconvenient and frustrating for tasks such as insertion of beams, because in general it requires the duplication of sub-tasks which have already been satisfactorily completed. For example if the error occurs during the location of the last note in a beamed cluster, then all notes in the cluster must be specified again. Alteration to the error handling procedures to correct this deficiency is straightforward. Nevertheless it is sometimes necessary to abort an edit task - for example when a command is incorrectly typed. Provision for this can be made in the modified error handling procedures by including a test for a special "abort task" character.

Secondly, many edit operations are repetitive in nature. For example, to change a stem direction to "up" at present the user must type "C,S,U" for every note which

is to be altered. It would be convenient to be able to type this command once, then to indicate in turn each note to be altered, for example by pressing the "RETURN" key when the current note is being pointed to. This procedure is used by the "I,BL" MOD edit command which inserts bar lines.

In addition to these points, the edit facilities should be extended to include a "horizontal shift for accidental" and a "horizontal shift for chord" command. Provision for the former has been made in the secondary note table, so that the implementation of these additional facilities is a straightforward extension of the present "C,X" command.

4.7 THE MUSIC PLOTTING SYSTEM GUTENBURG

The music plotting system GUTENBURG was developed by Howarth (1977a, 1977b) to overcome the limitations inherent in the EAI 640 display hardware. These limitations are due primarily to the comparatively small size of the storage oscilloscope, which contains 1024 by 800 addressable "dots" on a 21 cm by 16 cm screen. Thus the selection of the symbol size and stave line separation is a compromise between the need for a reasonable number of bars and stave groups on each page and the desire for elegant and detailed symbols. In addition to this fundamental limitation exist several hardware limitations which arise from non-linearities in the electron beam deflection and "dot write" system, and from drift within the controlling DACs. Thus, a dot which is nominally positioned at (X, Y) is actually positioned with a small error (which may be as large as several

resolution elements), and a line which is nominally straight may exhibit some curvature. For most applications these criticisms are irrelevant, but they are important for music typesetting because high aesthetic and visual quality is required. This point can be seen by comparing parts (d) and (e) of Figure 4.5.

GUTENBURG is essentially a FORTRAN IV transcription of the display system described in Section 4.5, and operates in batch rather than interactive mode. The hardware used is a 28 cm CALCOMP plotter which is controlled by a PDP 11/20 with 12K 16 bit words core memory. The PDP 11/20 is linked via 9600 bit per sec. lines to a Burroughs B6718 which has 160K 48 bit words of core storage. The music plotting system is executed by the B6718, while the PDP 11/20 controls the plotting of the primitive symbols. To improve the visual quality of the resulting typescript, the music is plotted at a size larger than is finally required, and photo-reduced to between 60% and 75% of the plotted size.

Full design and operational details are given by Howarth (1977a, 1977b), who also details the format of the paper tape which is punched by the interactive system and used as input to GUTENBURG.

4.8 EVALUATION AND EXAMPLES

In this section the process of music typescript production is illustrated by an example. Further examples are presented to indicate the kinds of results which can be achieved using the system.

The transcription, display, editing and final

hard-copy processes are illustrated in parts (a) to (e) of Figure 4.5. The example used is Chorale No. 6 by J.S. Bach.

The music as it is played on the organ keyboard is shown in MOD notation in Figure 4.5(a). Bar lines have been added (in this case using the joystick pointer) but no MOD performance editing has been carried out. The effect of the "roundoff" procedure is illustrated in Figure 4.5(b). This display was obtained using the commands "A" and "S" with the MOD display option set - normally the roundoff and transcription procedures are virtually transparent to the user, and the MOD display of the rounded music is suppressed. Figure 4.5(c) shows the TRAD display which is obtained after the rounded MOD note table is transcribed into the TRAD primary note table. This display is subsequently edited to produce the display of Figure 4.5(d). The typescript plotted by GUTENBURG is illustrated in Figure 4.5(e). No manual re-touching of this typescript has been done.

The total time taken to produce the edited display of Figure 4.5(d) from the start of the organ keyboard record process was 12.0 minutes.

The TRAD editing steps required in this example are summarised below:

TRAD Edit Step	Number of Operations
Change duration	6
Insert unison note	1
Insert rest	2
Insert note	1
Change note type (S,A,T,B)	4
Insert beam	11
TOTAL	25

The display times for the MOD notation examples (Figure 4.5(a) and (b)) are 3 seconds for each page. The TRAD notation displays (Figure 4.5(c) and (d)) required 5 seconds for each page. A large proportion of the TRAD display time is occupied performing core-image phase overlays from disc storage (six phase overlays are required in this example - see Figure 4.4 and Table 3.5).

Further examples of music typescript produced by the system are given in Figures 4.6 and 4.7. The first is Mozart's "Minuet from Don Juan" (arranged for piano), while the second is "Interface 640", a trio for oboe, clarinet and bassoon by Philip Norman. The latter deserves special mention because it was commissioned specifically as a "trial run" to test and evaluate the system described herein (Howarth, 1977a). Titles and text in this example were added using "Letraset" (a "stick-on" labelling device), and phrase marks etc., were drawn manually.

4.9 SUGGESTIONS FOR FURTHER DEVELOPMENT

The music typescript production system has been extensively evaluated, and Howarth (1977a) gives a detailed discussion of those aspects of the system for which refinement or further development is desirable. Many of these points relate to the user convenience of the TRAD edit facilities and have already been mentioned in Section 4.5. Other aspects are a reflection of the present state of the system development, and these are now considered.

The desirability of inserting rests and beams automatically during the transcription process (rather than manually during the edit stage) has been noted (Section 4.4). At present these two aspects account for most of the total editing time, so that automatic rest and beam insertion would significantly reduce the overall amount of editing required. The manner in which this could be achieved is outlined in Section 4.4.

The usefulness of automatic four part harmony formatting in reducing the number of typescript edit operations is self evident. A further display format is required to accommodate piano music - a possible method of implementing this is described in Section 4.5.3.

The development of score format display facilities is considered to be important. A score is music copy which shows all instrumental parts separately (one line of music on each part) but with all parts shown on each page. This could be achieved without requiring a drastic re-organisation of the system, since separate primary and secondary note tables for each part could be constructed as at present. These tables could then be displayed "in parallel" in score format by a suitably modified display control procedure. The basic stave group layout on the page also needs to be adjusted appropriately.

At present tie and slur symbols and phrase indications must be added manually to the final hard copy. Such "symbols" could be incorporated into the display system by requiring the user to specify (as an edit task) the beginning and end points of the curved line symbol and its

direction of curvature. The symbol could then be drawn as a segment of a circle or elipse. A further refinement would permit the user to specify intermediate points through which the curved line passes.

Text annotation should be extended to include musical annotative symbols as well as alphanumeric characters. Such symbols include staccato dots, accent marks, pause marks, fingerings and bowing indications (for string players), breath marks (for wind players) etc. It is also important that annotative messages be able to be specified as "absolute" (at a defined point on the page) or "relative" (to a specified note or bar line). This point is discussed in Section 4.3.4.

The effect of hardware limitations upon the visual quality of the final hard copy have already been mentioned (Section 4.7). Kassler (1977) discusses the production of music typescript by both graphical plotters (such as the one used by GUTENBURG) and by special photo-typesetters. Photo-typesetting is now extensively used in the text printing industry, and is usually performed under (special-purpose) computer control so that justification is performed automatically. The method of operation is as follows: A special disc or drum incorporates transparencies of each character in a variety of fonts. The disc rotates between a controllable light source and a photo-sensitive paper which will become the master copy. As the selected character on the disc passes in front of the light source, the latter is pulsed. A light beam focusing and positioning system causes the character to be exposed momentarily on the

photo-sensitive paper. The character "write position" is then shifted and the next required character is selected and exposed similarly. The character size is adjusted by suitably controlling a magnifying lens in the light beam path.

The author's observation of text photo-typesetters has led to the conclusion that they are significantly faster than graphical plotters and that they produce a sharper, more clearly defined image. Kassler confirms this conclusion and observes that special-purpose music symbol discs are available commercially. Kassler also comments that the use of a photo-typesetter is advantageous in a commercial environment because the one machine can be used to produce both text and music. Naturally, in the University (experimental) environment the purchase of photo-typesetting hardware is neither practicable nor possible for a project such as this. However, should the music type-setting system be further developed for a commercial environment the advantages offered by a photo-typesetter system should be carefully considered.

TABLE 4.1

TRADITIONAL NOTATION PRIMARY DATA TABLE
DATA STRUCTURE SYNTAX IN BACKUS NAUR FORM

(a) TERMINALS GIVEN IN MNEMONIC FORM

```
<digit>                 ::=  0/1/2/3/4/5/6/7/8/9
<integer>               ::=  <digit> / <integer> <digit>
<pitch>                 ::=  <integer>
<duration>              ::=  <integer>
<minus duration>        ::=  -<integer>
<basic note>            ::=  <pitch> <duration>
<note>                  ::=  <basic note> /
                             <accidental> <basic note> /
                             <note prefix> <note>
<rest>                  ::=  <pitch> <minus duration>
<accidental>            ::=  SHARP / FLAT / NATURAL /
                             DOUBLE-SHARP / DOUBLE-FLAT
<note prefix>           ::=  <beam operator> / <tie operator> /
                             <split operator>
<beam operator>         ::=  BEAM-START <cluster identifier> /
                             BEAM-MIDDLE <cluster identifier> /
                             BEAM-END <cluster identifier>
<cluster identifier>    ::=  <integer>
<tie operator>          ::=  TIE-START <cluster identifier> /
                             TIE-MIDDLE <cluster identifier> /
                             TIE-END <cluster identifier>
<split operator>        ::=  SPLIT-START / SPLIT-END
<bar line>              ::=  BARLINE <duration>
<shift operator>        ::=  SHIFT-OP <duration>
```

TABLE 4.1 Continued

```
<vertical cluster>          ::=  <note> / <rest> /
                                 <vertical cluster> <note> /
                                 <vertical cluster> <rest>
<vertical cluster string>   ::=  <vertical cluster> /
                                 <vertical cluster>
                                 <shift operator>
                                 <vertical cluster>
<bar>                       ::=  <vertical cluster string>
                                 <bar line>
<primary data table>        ::=  <bar line> <bar> /
                                 <primary data table> <bar>
```

(b) TERMINAL CODE VALUES

SHARP	=	3 000
FLAT	=	3 010
DOUBLE-SHARP	=	3 020
DOUBLE-FLAT	=	3 030
NATURAL	=	3 040
BEAM-START	=	2 000
BEAM-MIDDLE	=	2 100
BEAM-END	=	2 200
TIE-START	=	1 600
TIE-MIDDLE	=	1 610
TIE-END	=	1 620
SPLIT-START	=	1 500
SPLIT-END	=	1 520
BARLINE	=	10 000
SHIFT-OP	=	1 000

TABLE 4.2

TRAD PRIMARY NOTE TABLE DURATION CODE

Mnemonic	Name	Value	Code
SQ	Semiquaver	1/16	1
Q	Quaver	1/8	2
QD	Quaver dot	1/8 + 1/16	3
C	Crotchet	1/4	4
	Illegal		5
CD	Crotchet dot	1/4 + 1/8	6
CDD	Crotchet double dot	1/4 + 1/8 + 1/16	7
M	Minim	1/2	8
	Illegal		9
	Illegal		10
	Illegal		11
MD	Minim dot	1/2 + 1/4	12
	Illegal		13
MDD	Minim double dot	1/2 + 1/4 + 1/8	14
	Illegal		15
SB	Semibreve	1	16

Illegal entries correspond to tied combinations of notes which must be coded accordingly in the primary note table.

PRIMARY NOTE TABLE

10000	0	bar line
59	8	note
56	8	
51	8	
32	6	
1000	6	shift operator
39	2	
1000	2	
51	-2	rest
44	8	
1000	2	
2000	1	link start operator
58	2	
1000	2	
2100	1	link middle operator
56	2	
1000	2	
2200	1	link end operator
54	2	
10000	2	
3000	0	accidental
53	8	
37	8	
10000	8	

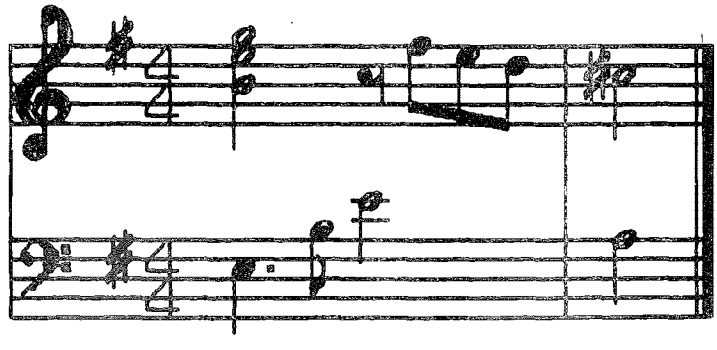


Figure 4.1 An illustrated example of the music encoding data structure used in the system (see also Table 4.1).

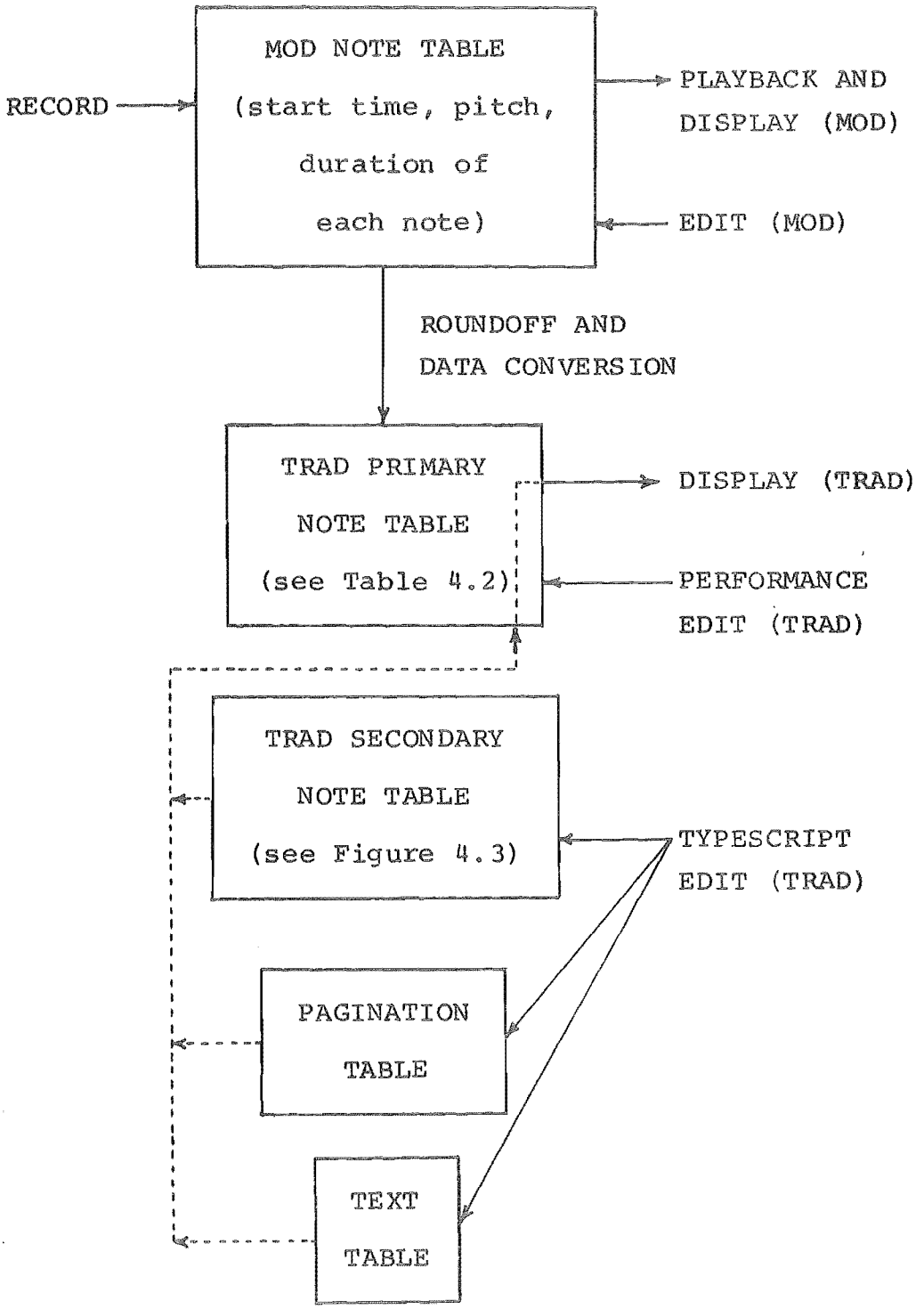
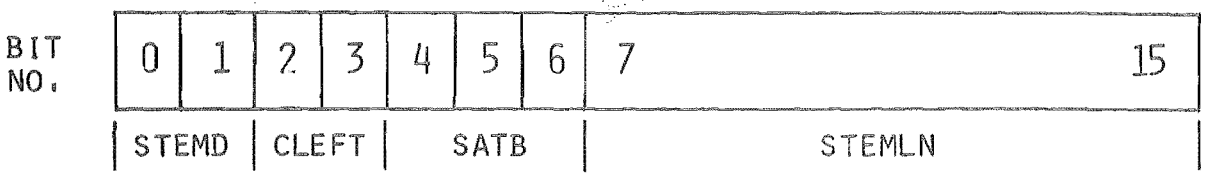


Figure 4.2 Summary of the data structure and the interrelationships between the tables.

WORD 2



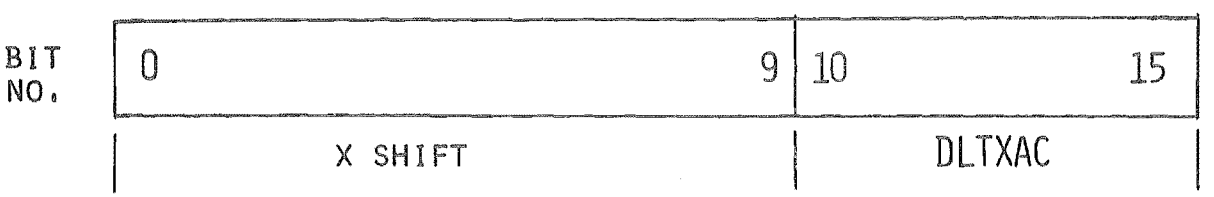
STEMD = 0 Stem direction to be decided by algorithm
1 Stem up
2 Stem Down

CLEFT = 0 Note staff type to be decided by algorithm
1 Note in Treble staff
2 Note in Bass staff

SATB = 0 Note type to be decided by algorithm
1 Note in Treble and Stem Up
2 " " " " " Down
3 " " Bass " " Up
4 " " " " " Down

STEMLN = Length of Stem (in 1/100 inch)

WORD 3



XSHIFT = Magnitude of Note Shift (horizontal)

DLTXAC = Magnitude of Accidental Shift (horizontal)

Notes: (1) Each entry (in Secondary Note Table) contains three words.

(2) WORD 1 is a pointer to the edited note entry in LIST2.

(3) 16 bit word convention applicable to the EAI 640 Digital Computer.

Figure 4.3 Format of Secondary Note Table.

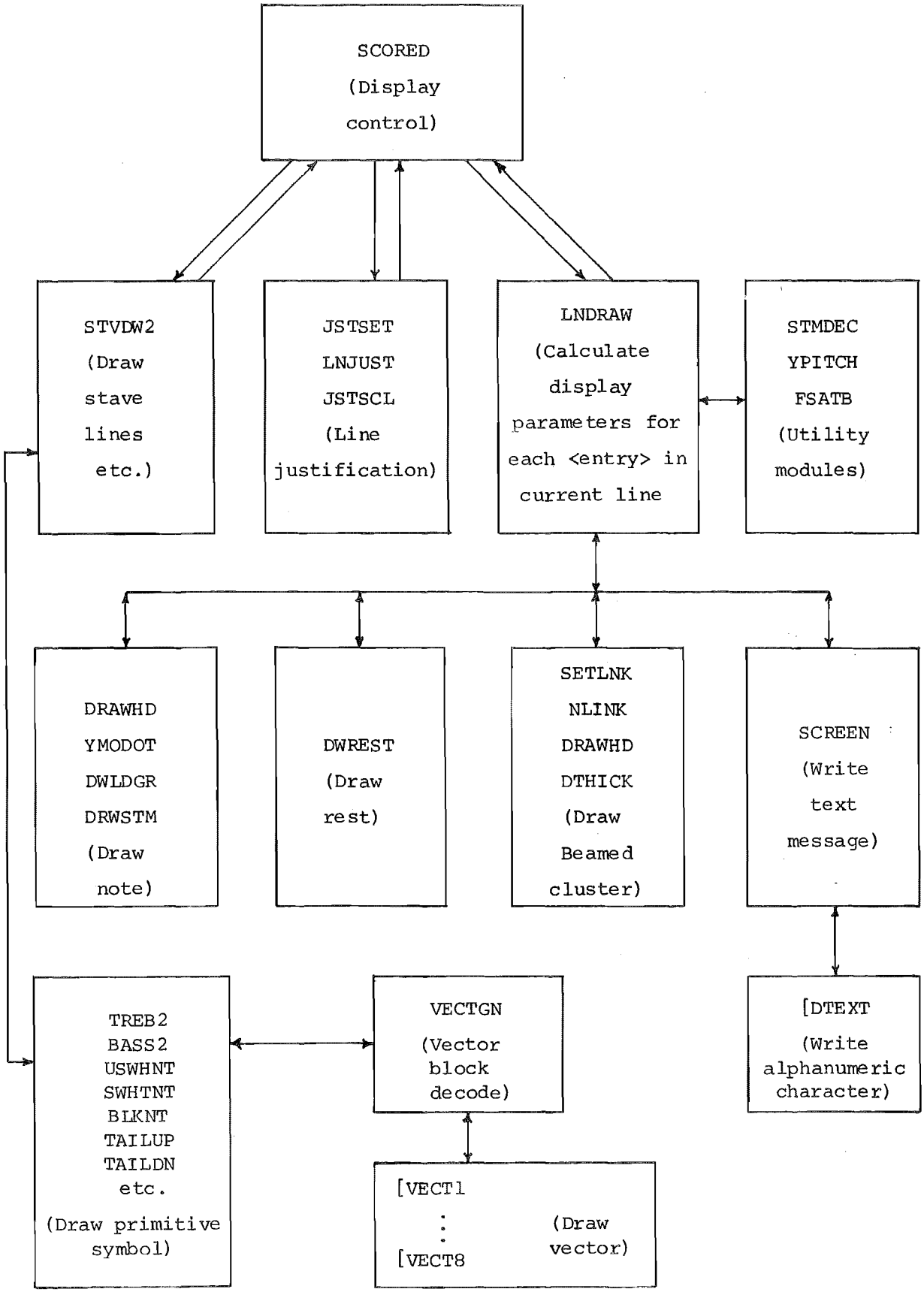


Figure 4.4 Organisation of the TRAD display modules.
(See also Table 3.5)

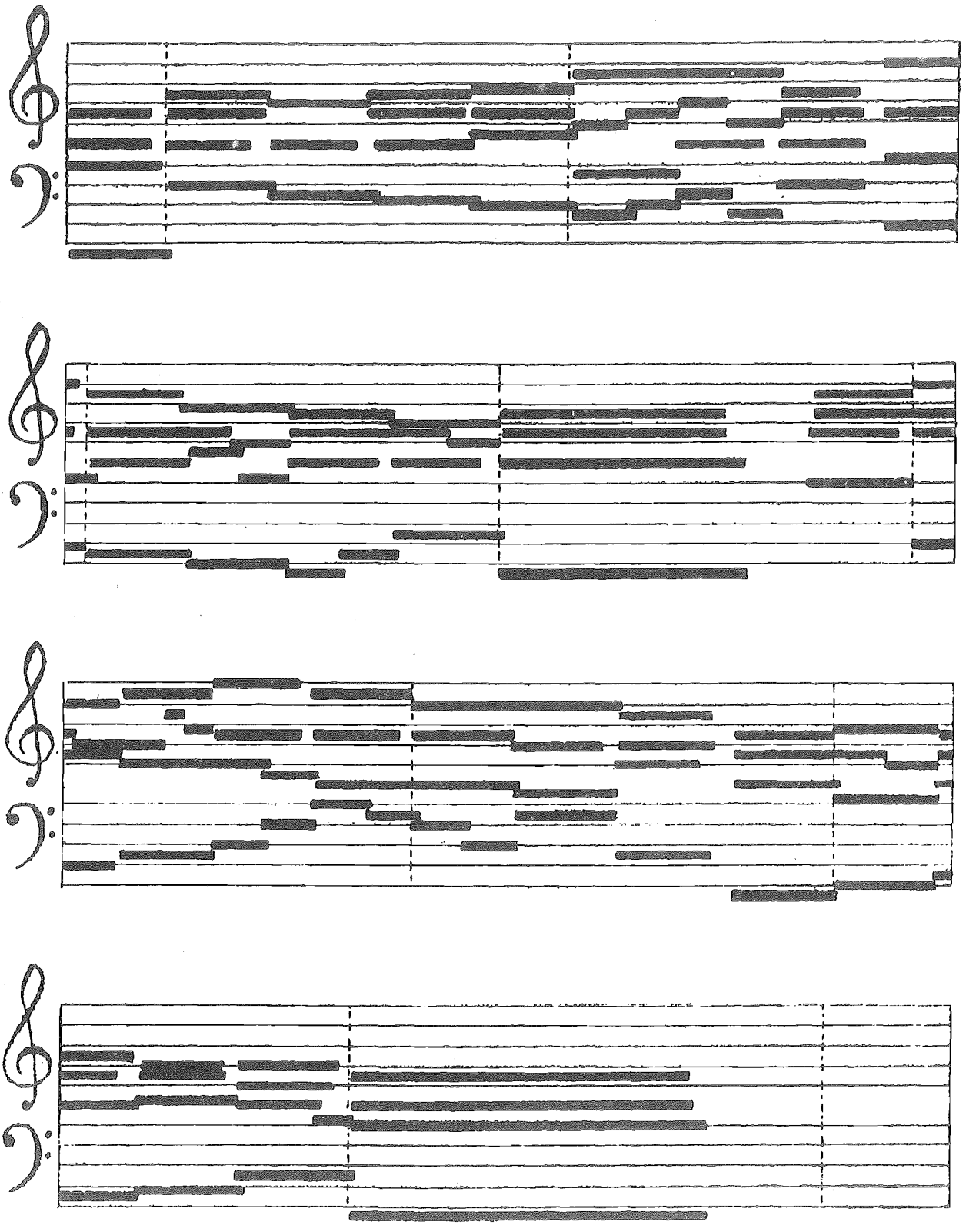


Figure 4.5 Illustrating the TRAD transcription and editing procedure. The example shown is Chorale Number 6 by J.S. Bach.

(a) Display in MOD notation of music as played on the organ keyboard. Bar lines have been added, but no editing has been done.

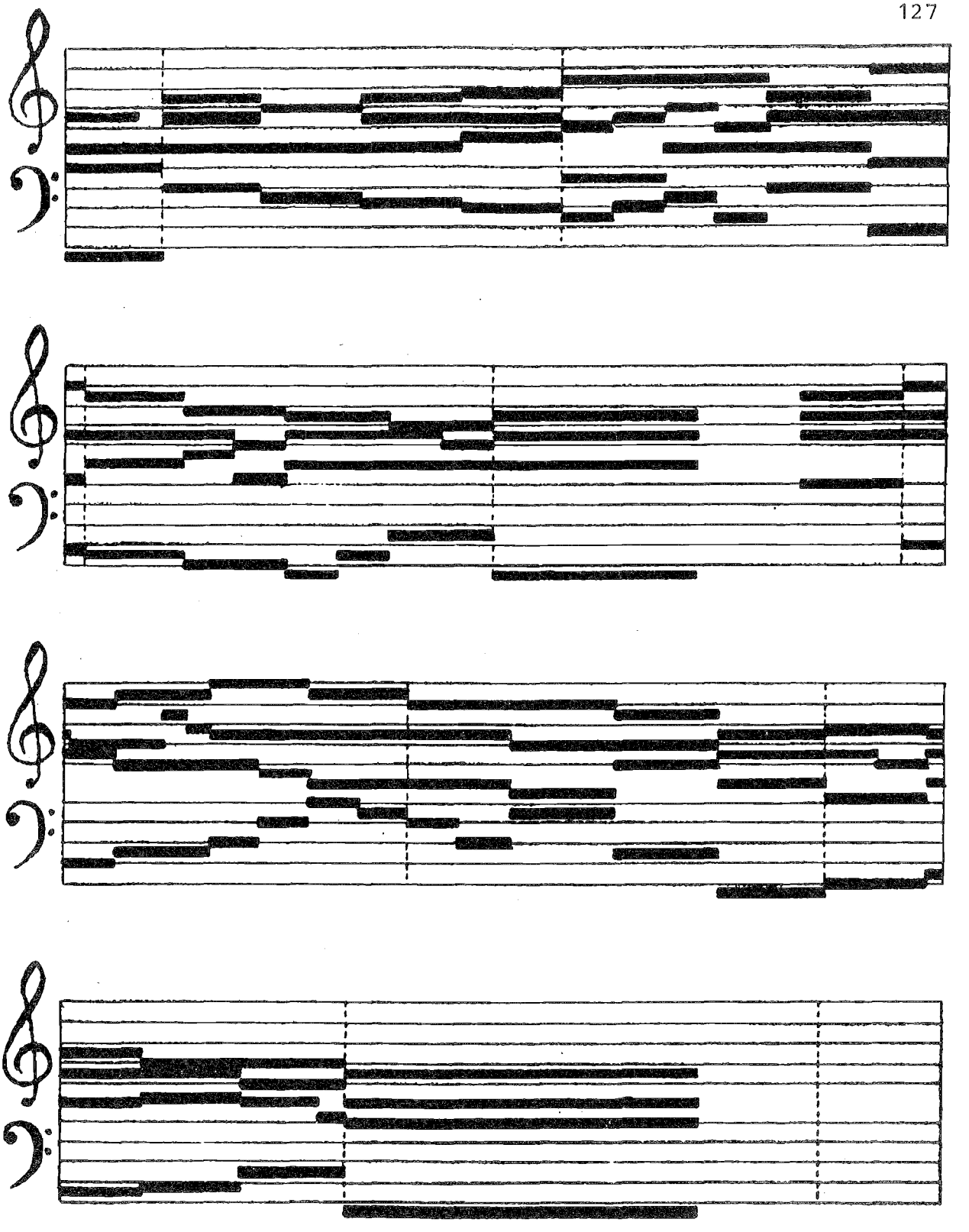


Figure 4.5 (Continued).

(b) Display in MOD notation of the music shown in (a) after roundoff. Observe that the start and end time of each note is aligned to the appropriate "time slot boundary".

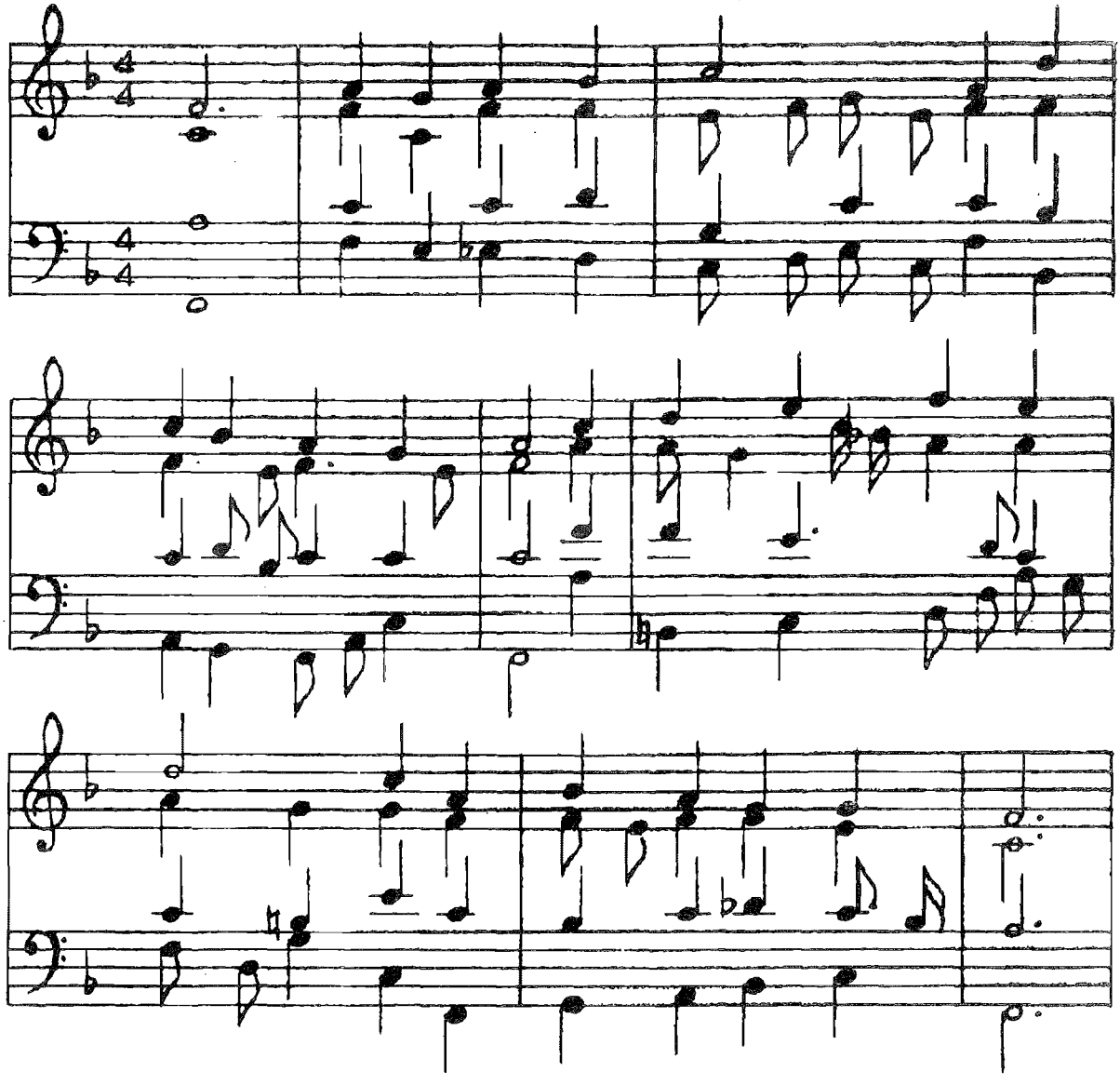


Figure 4.5 (Continued).

(c) Display in TRAD notation of the music shown in (b) after transcription from MOD to TRAD data form. The procedure so far requires no manual intervention.

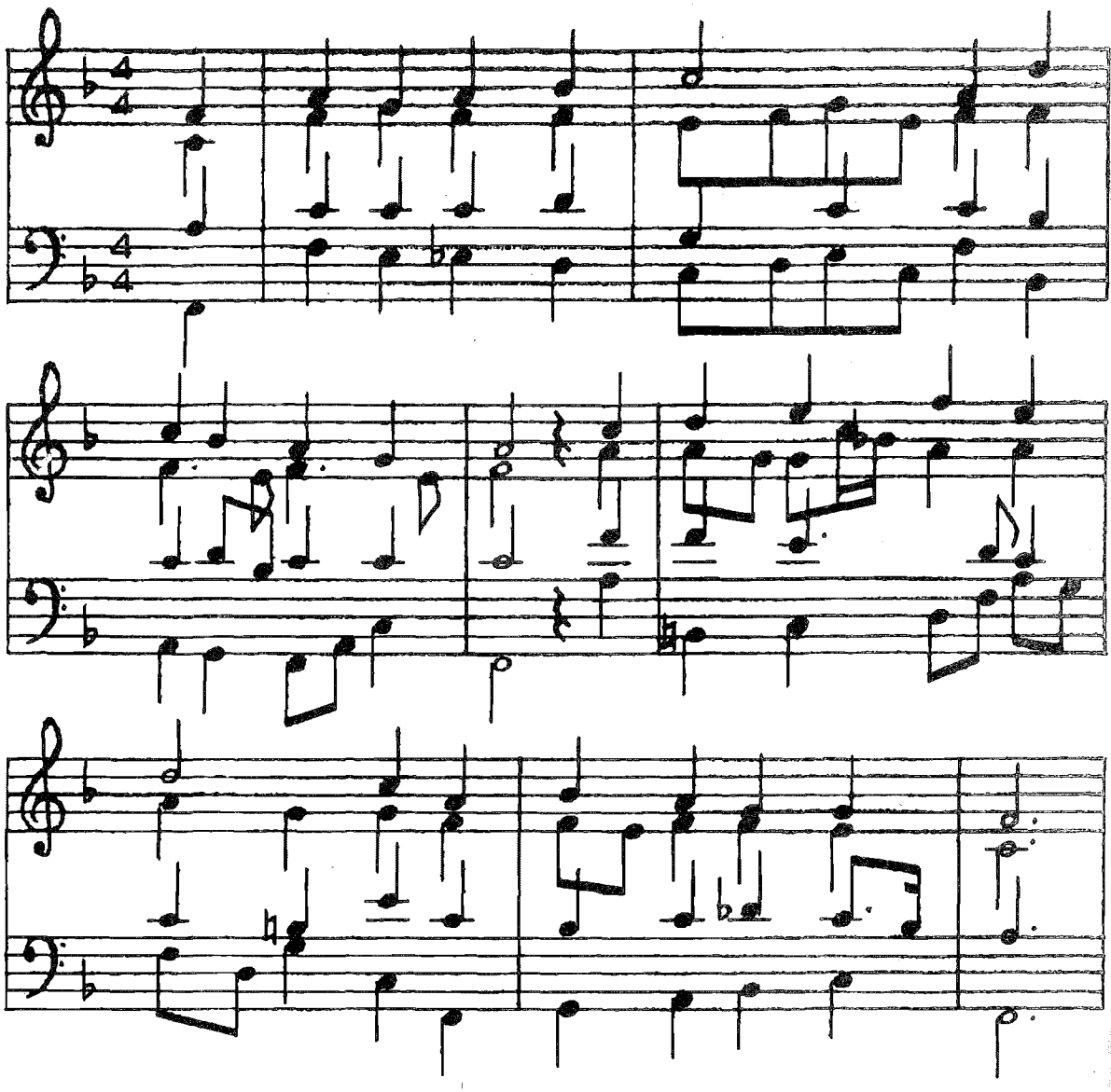


Figure 4.5 (Continued).

(d) Display in TRAD notation of the music shown in (c) after manual editing has been carried out. The total elapsed time between the start of recording and the completion of this display was 12.0 minutes.

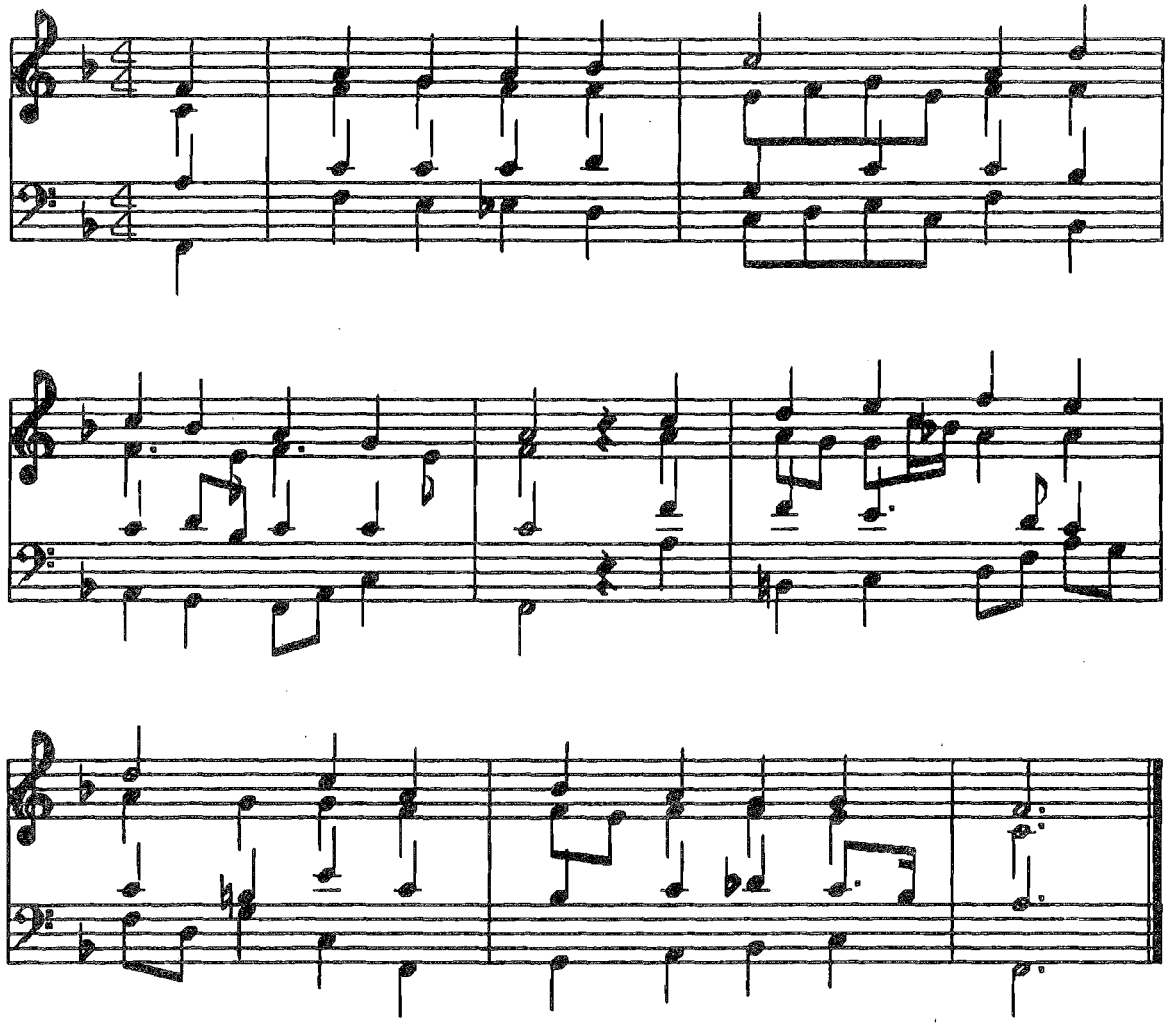


Figure 4.5 (Continued).

(e) The display shown in (d) after plotting by GUTENBURG. The limitations of the storage oscilloscope for producing elegant music copy are evident if parts (d) and (e) are compared.

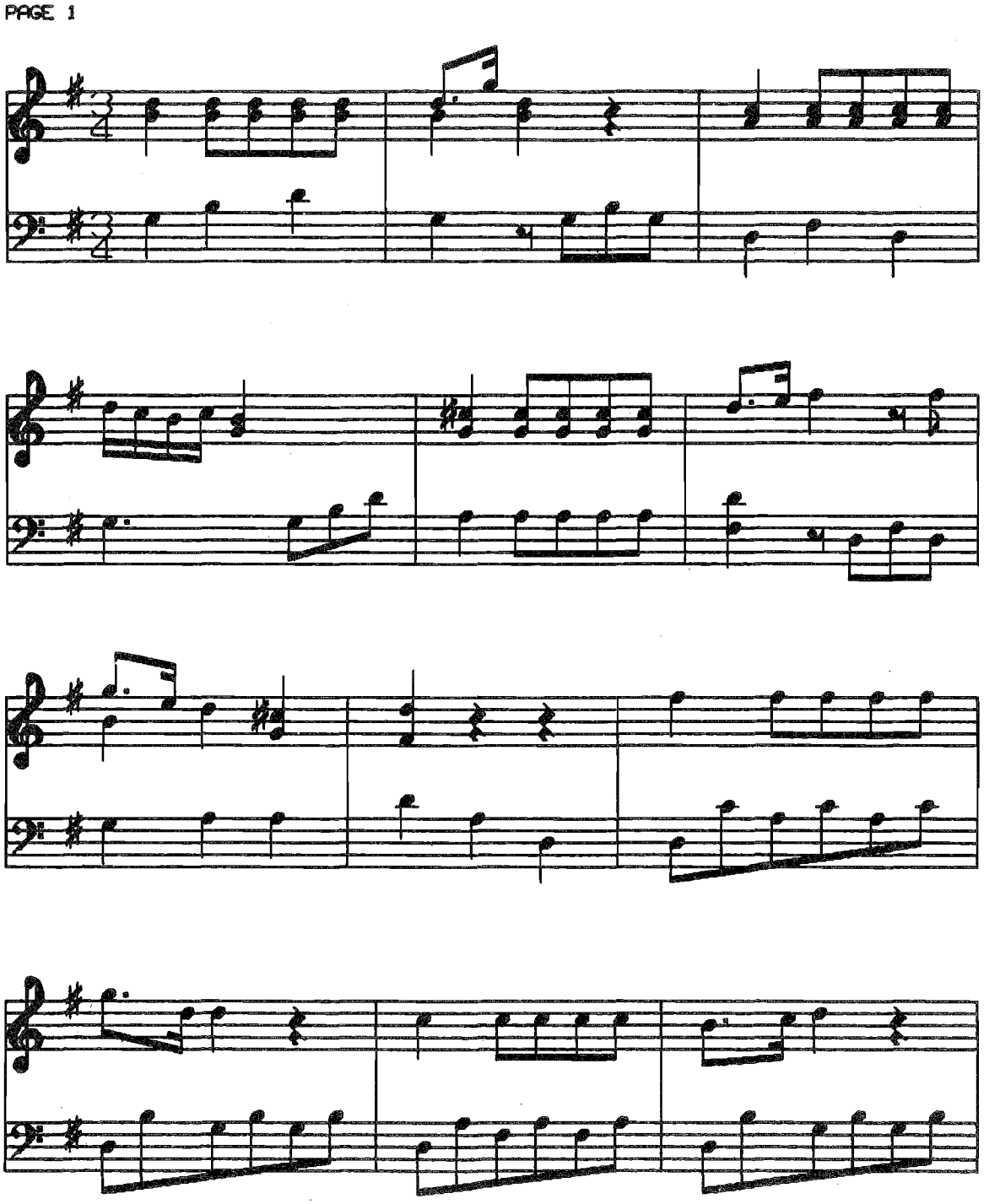


Figure 4.6 "Minuet, from Don Juan", (Mozart).
Final copy after batch processing (2 pages).

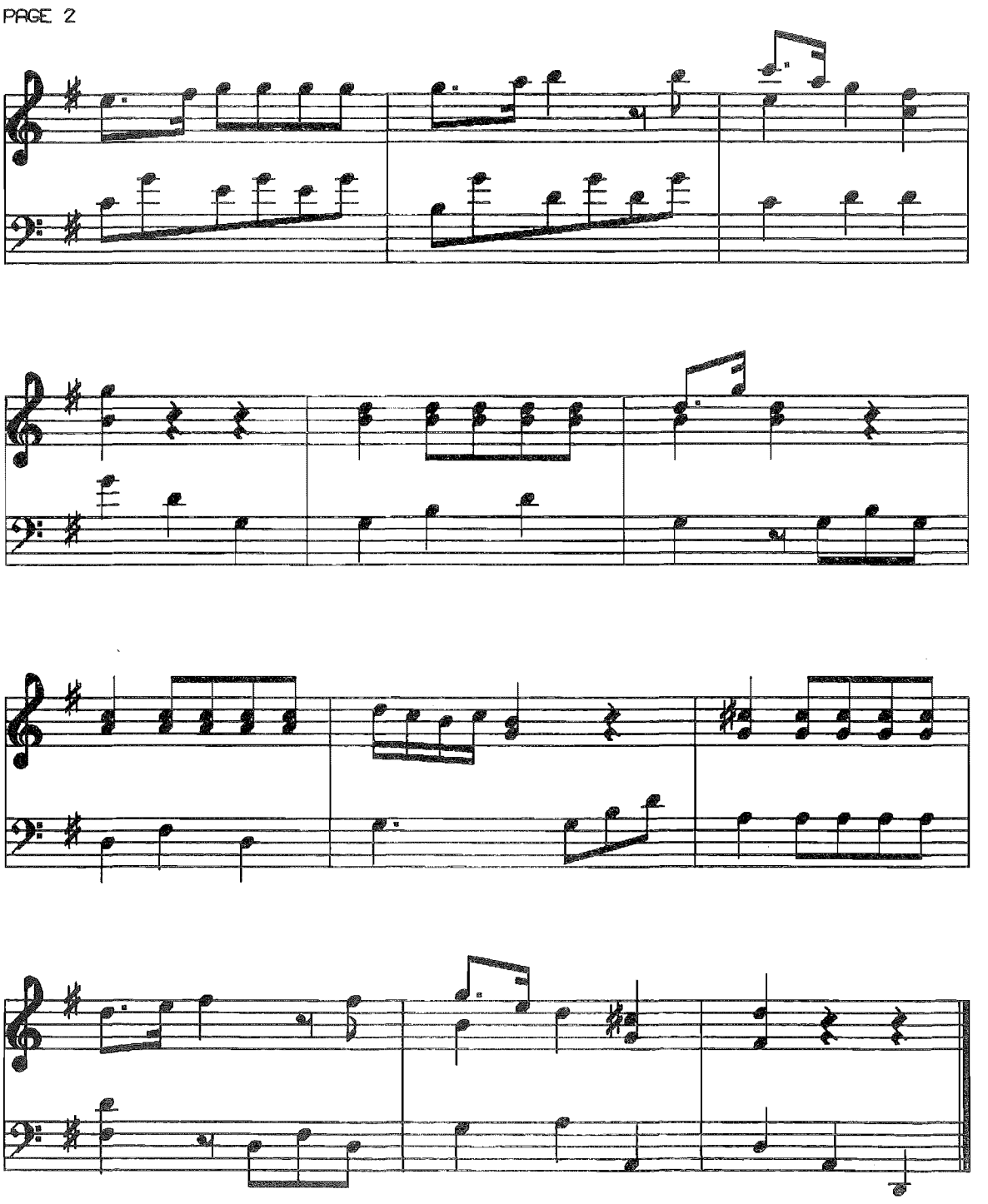


Figure 4.6 (Continued).

INTERFACE 640

Oboe

Philip Norman

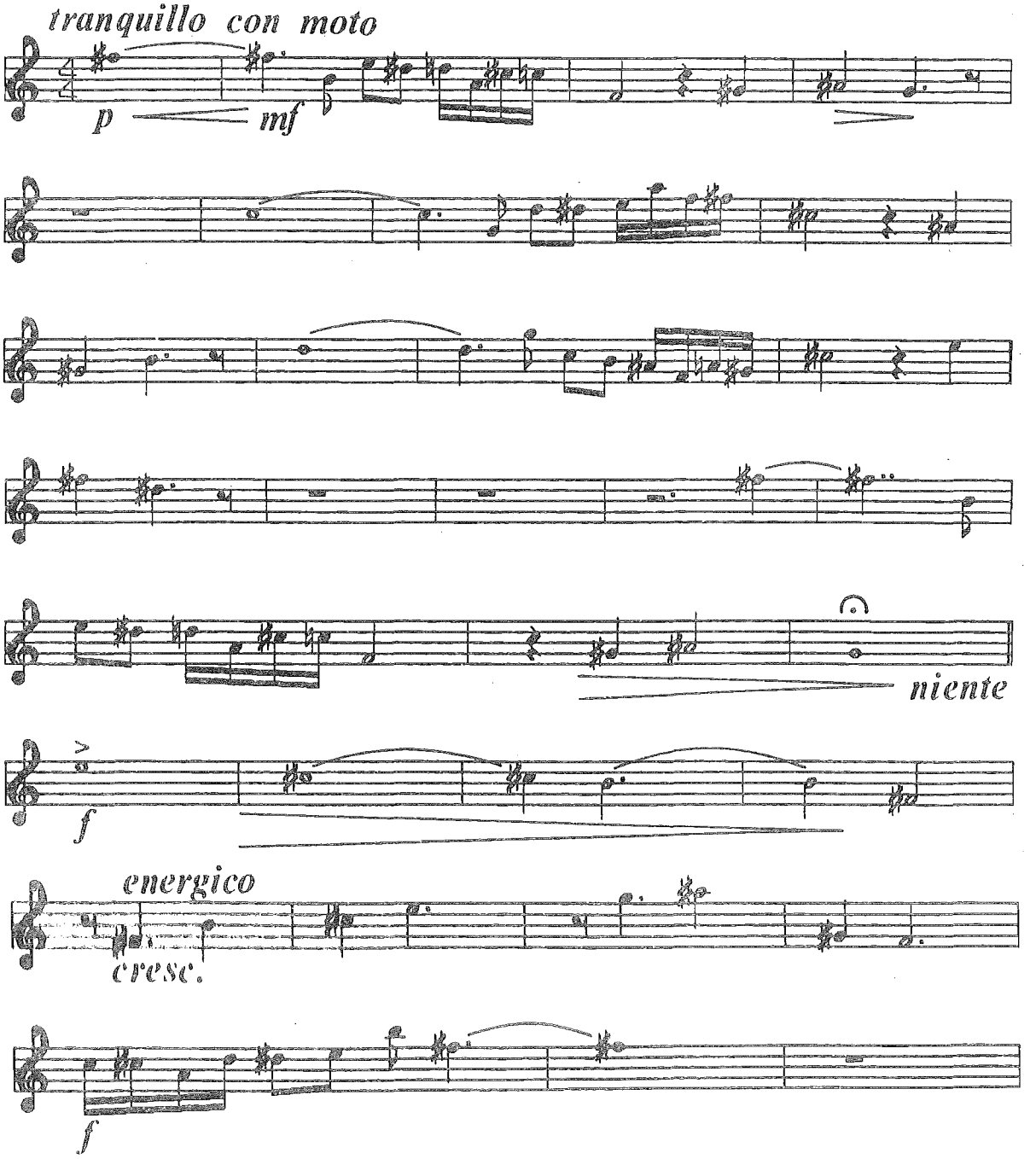


Figure 4.7 "INTERFACE 640" - Oboe part (2 pages).

Oboe

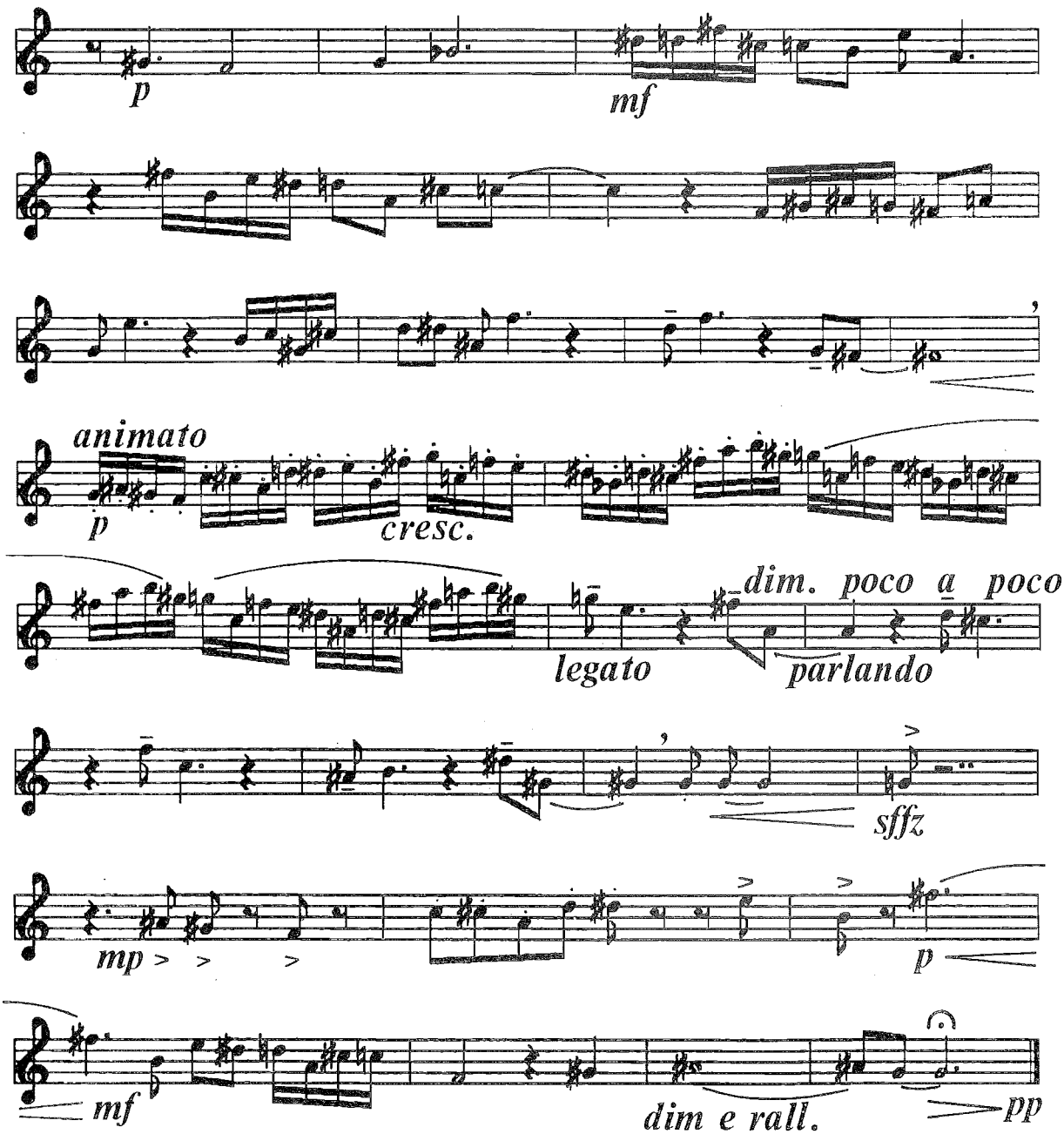


Figure 4.7 (Continued).

INTERFACE 640

Clarinet

Philip Norman

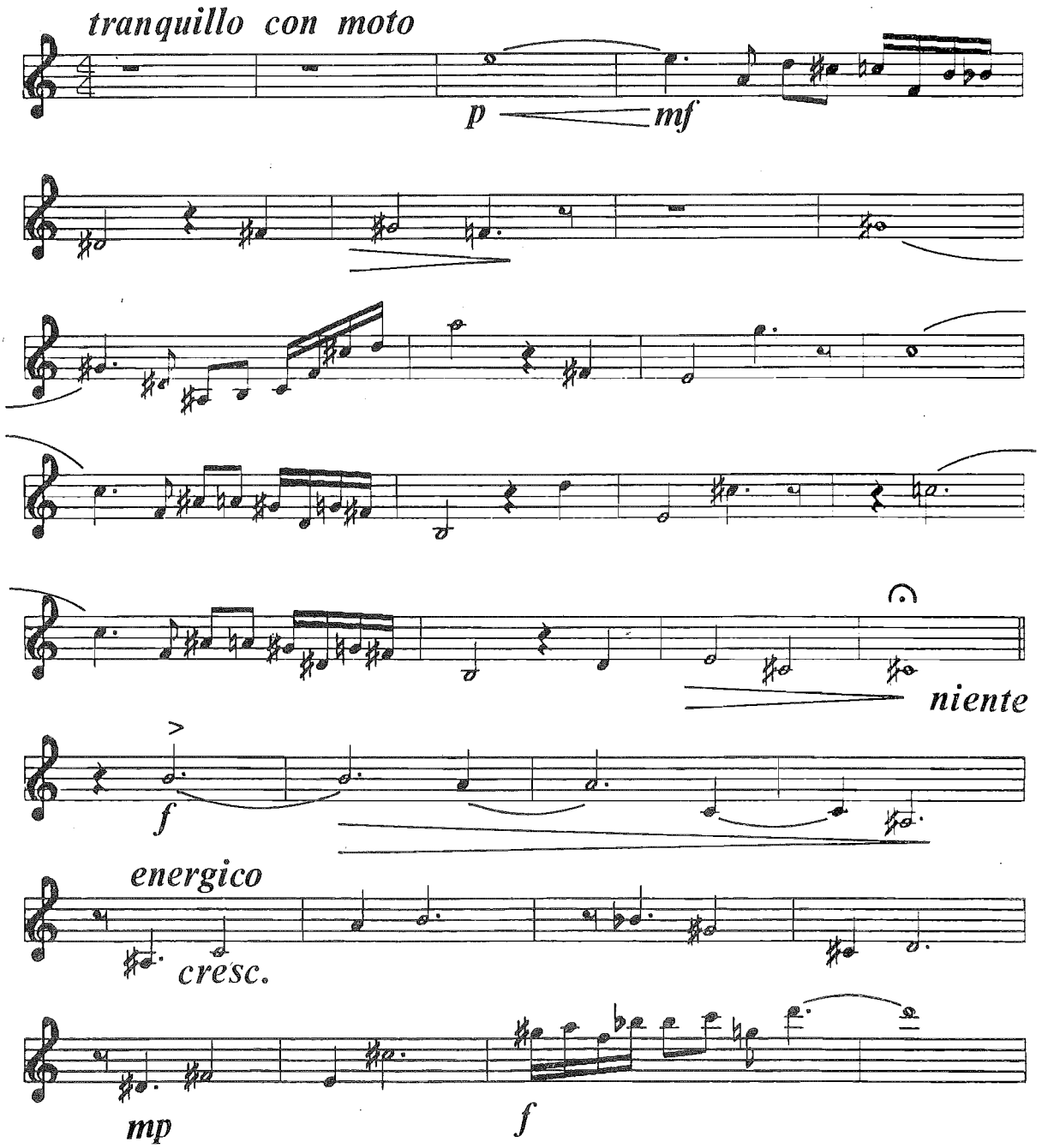


Figure 4.7 "INTERFACE 640" - Clarinet part (2 pages).

Clarinet

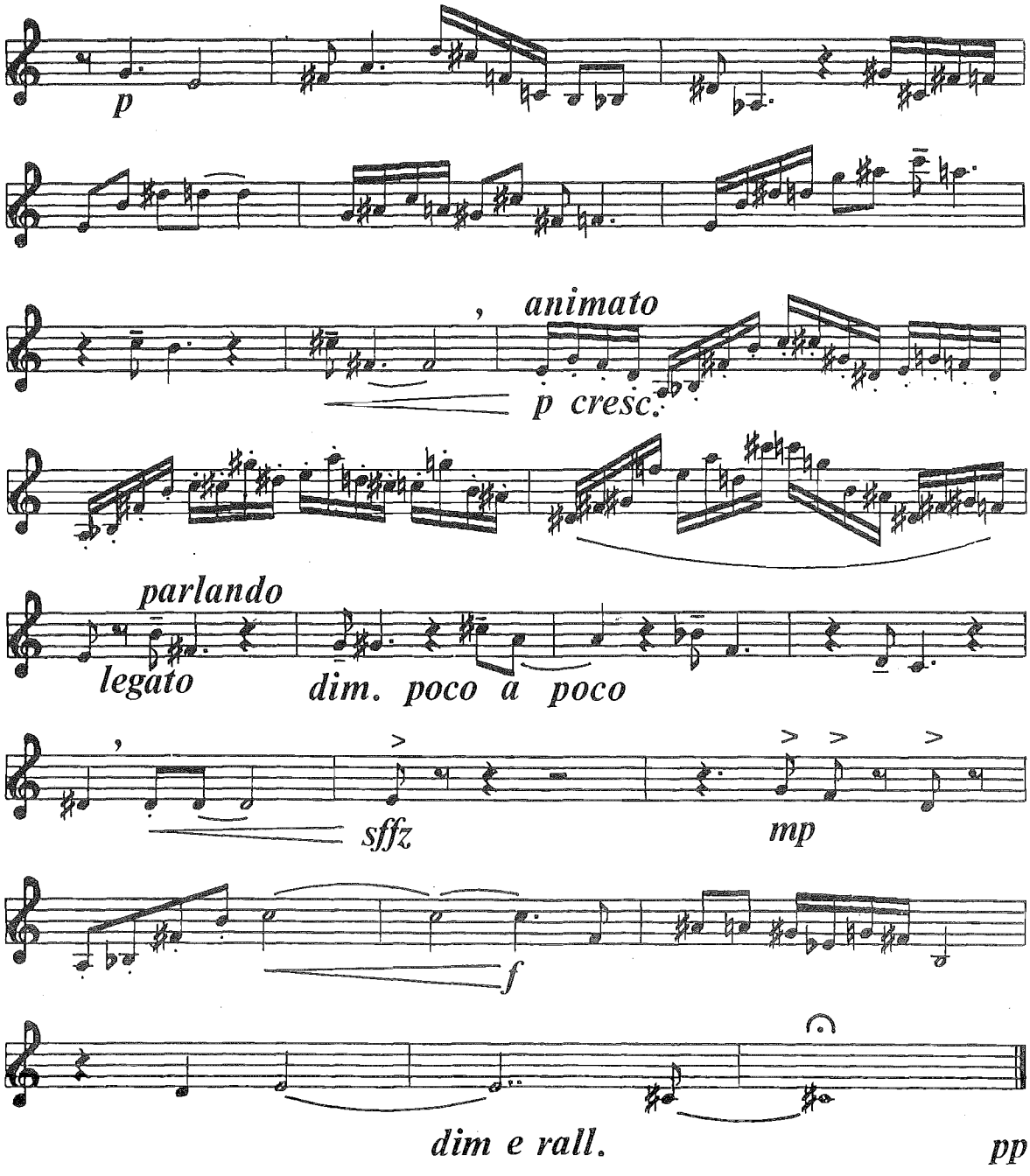


Figure 4.7 (Continued).

INTERFACE 640

Bassoon

Philip Normar

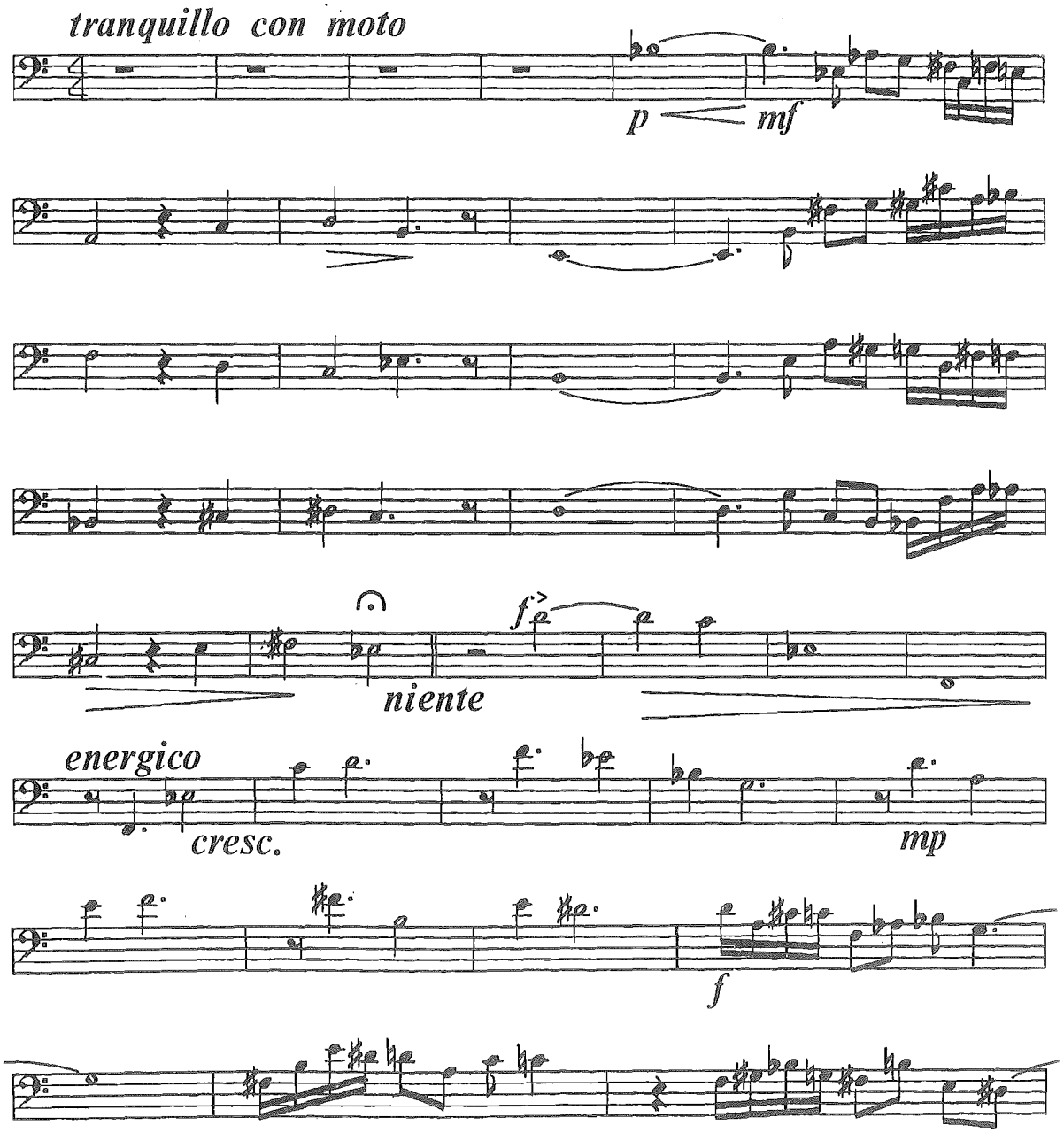


Figure 4.7 "INTERFACE 640" - Bassoon part (2 pages).

Bassoon

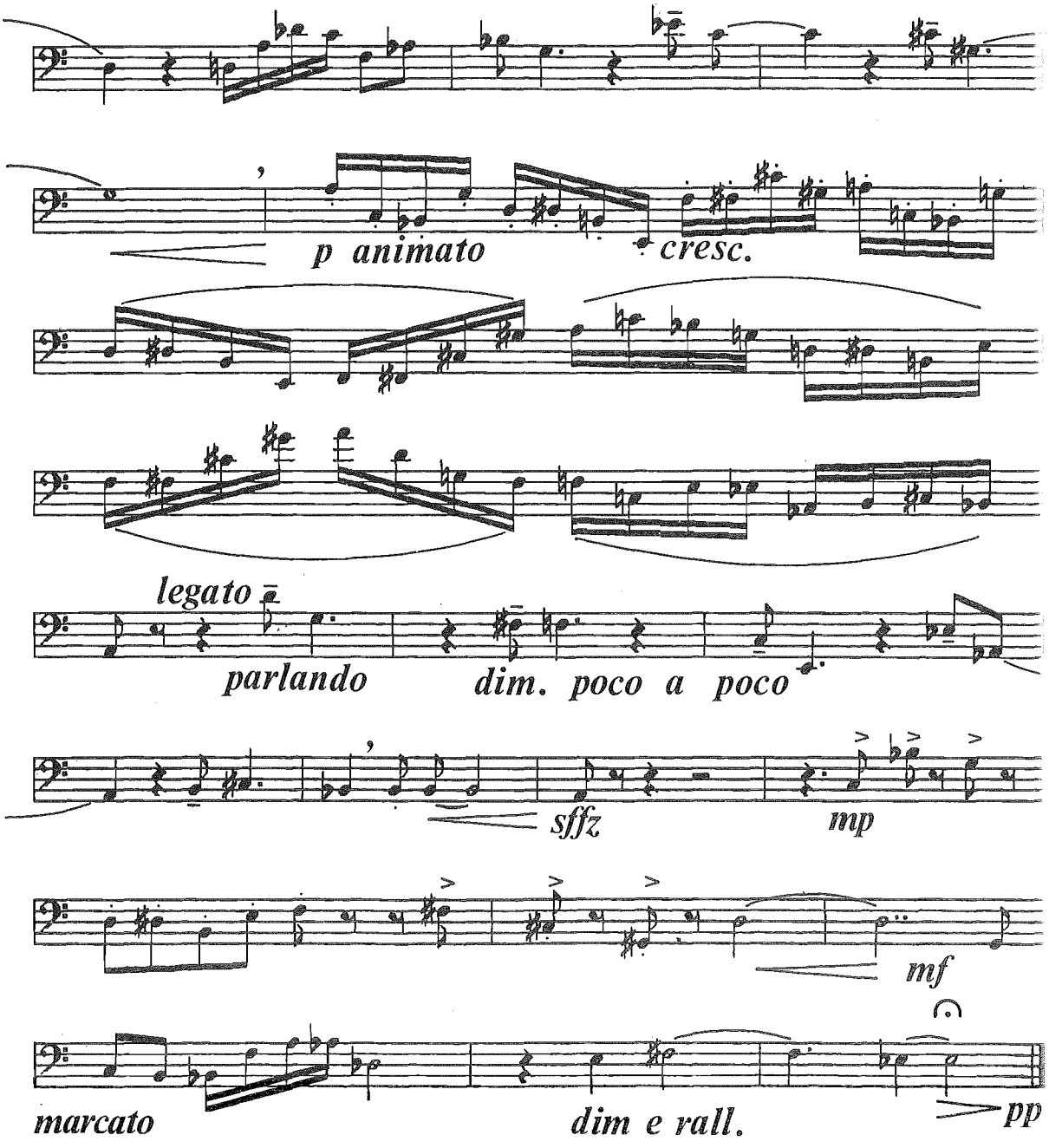


Figure 4.7 (Continued).

PART 2

ELECTRONIC SOUND SYNTHESIS

CHAPTER 5

A REVIEW OF ELECTRONIC SOUND SYNTHESIS TECHNIQUES

5.1 INTRODUCTION

The use of electronic rather than acoustic techniques for the generation of sounds greatly extends the range of sound timbres available to musicians and composers. In addition to providing the means for production of new sounds, electronic techniques permit enhanced control over existing sounds, so that complex new timbres can be formulated and performed without mechanical limitations to virtuosity.

The application of electronic techniques to music performance and sound synthesis has already been mentioned in Section 2.2. There, the distinction is made between the parameters which are required to specify a performance and those required to specify a sound timbre. This chapter considers in detail the use of electronic techniques for the synthesis of sounds. Both analogue and digital techniques are discussed although the emphasis is on the latter. Some of the methods described here are also applicable to speech (see Section 7.3 for a discussion of the similarities and differences between speech and music). Additionally, the problems of specifying and controlling the timbral characteristics of the synthesised sounds are considered.

In this context, "timbre" refers to the time-varying spectral structure of a sound. The problem of determining those particular spectral characteristics which permit the recognition and imitation of musical instruments is not considered here, despite its relevance. For a discussion of this topic, see for example Eagleson and Eagleson (1947), Richardson (1954), Luce (1963), Lehman (1964), Blackman (1965), Luce and Clark (1965, 1967), Beauchamp (1967), Freedman (1967), Strong and Clark (1967a, 1967b), Keeler (1972) and Robson (1976). The thesis by Grey (1975) is also relevant.

It is worth commenting here that many of the problems encountered by composers who use electronic synthesisers to produce "electronic music" relate to operational difficulties (i.e. inadequate control facilities) rather than to limitations inherent in the synthesis technique. As Howe (1972) observes, a synthesiser limits the music which may be conceived with it by the operational skills necessary to make it respond. Howe complains that "many of the devices on some synthesisers have controls that are more appropriate for electrical test equipment than for a musical instrument", and that for many synthesisers "you need more than two hands to do everything you want to". The need for adequate yet operationally simple control facilities is now widely recognised (cf. Covell, Holmes and Kabowiak, 1971; Gross and Leibig, 1976; also Section 6.1). However, the difficulties encountered are not easily overcome and much work remains to be done.

5.2 ANALOGUE TECHNIQUES

Analogue techniques for electronic sound synthesis fall into two broad categories - those which start with acoustic signals (the so-called "concrete sounds") and perform extensive electronic processing such as amplification, filtering and modulation, and those which use "pure" signals of electrical origin such as the output of an oscillator.

5.2.1 Early Instruments

Historically, the origins of electronic music are generally linked with Edison's invention of the phonograph in 1877 (Backus, 1970), although sound recording and reproduction remained a mechanical rather than electrical process until the development of "Orthophonic" recording in 1925 by J.P. Maxfield and H.C. Harrison (cf. Fletcher, 1929). The work of A.G. Bell, C.S. Tainter and E. Gray in the field of sound transmission and recording in the mid 1870's and early 1880's should also be mentioned here (cf. Bowles, 1970).

The earliest instrument to produce musical sounds electrically seems to be the "Telharmonium", which was patented by T. Cahill in 1897 (Taylor, 1965). This device uses an assembly of rotary generators which produce sinusoidal tones of various frequencies and intensities. Switches permit the synthesis of arbitrary spectra (using additive synthesis), and a volume control provides dynamic variation. The Telharmonium was intended to produce music capable of transmission over telephone lines, but was never

developed beyond the demonstration stage. Nevertheless, it is regarded as the ancestor of the electronic synthesiser (Prieberg, 1960).

The development of electronic technology following L. De Forest's introduction of the audion in 1907 led to numerous electronic instruments which use oscillators, modulators, filters and amplifiers. Lewer (1948) and Prieberg (1960) describe these early instruments, typical of which is the "Aetherophone" (later called the "Theremin"). The aetherophone was developed by L. Theremin in 1924, and consists of two radio-frequency oscillators which produce beat notes. The pitch of the resulting sound is altered by varying the distance between a wire-loop aerial and a metal rod which is held by the operator. A similar but more advanced electronic instrument was produced by M. Martenot in 1928 (cf. Bowles, 1970). Called the "Ondes Musicales" (later the "Ondes Martenot"), the instrument also uses two radio-frequency oscillators to produce a beat oscillation by heterodyning. Pitch is controlled using a moveable electrode. The composers Dutilleux, Honegger, Messiaen, Milhaud and Varèse have all written music for the Ondes Martenot (Raven-Hart, 1930).

5.2.2 The Electronic Organ

The electronic organ was first developed in the early 1930's as an imitative instrument (rather than one designed to stimulate the imagination of avant-garde composers), and has become an accepted instrument in its own right. The Compton electrone, patented in 1932 by Bourn and Compton (Taylor, 1965), uses an electrostatic technique for the

production of a repetitive waveshape. In its basic form it incorporates twelve identical tone-generating units, each producing seven notes spaced at octave intervals. A driving motor turns each generator with a belt and pulley mechanism. The ratios of the generator pulley diameters are chosen so that the generator speeds are in the correct ratios to produce the twelve notes of the equally tempered scale. In this manner a total of eighty-four notes are produced by the twelve tone generator units. The tone generator mechanism consists of an engraved rotor/stator system, with capacitive coupling controlled by polarising voltages. The shape (and hence timbre) of the repetitive signal produced depends on the "waveshape" which is engraved on the stator. Tonal quality may be further controlled by adding proportions of the outputs of several tone generators. Vibrato effects are produced by an oscillating auxiliary pulley which imposes fluctuations on the belt drive speed. Envelope effects such as attack and decay are produced electronically by analogue networks.

The Hammond organ was introduced in 1934 (Taylor, 1965) and uses electromagnetic tone generation. The basic oscillator element is a steel wheel with profiled teeth. As the tone wheel rotates in a localised magnetic field, a sensing coil responds to field fluctuations caused by the time-varying steel profile. A total of ninety-six wheels are used, and are arranged in twelve groups of eight. Each of the twelve groups is rotated at constant speed, with the speed ratios chosen to produce the frequencies of the equally-tempered scale. The number of teeth on the

tone wheels of each group differ by multiples of two, so that successive octaves are produced. As in the Compton system, tone colouring is achieved by the addition of harmonics in controllable proportions. Tremolo is implemented using an electromechanical gain control which is driven by an eccentric wheel.

Both the electromechanical organ systems described above have been superseded by systems which use electronic methods for frequency generation. An early example is the Wurlitzer organ, which uses electronic oscillators as the primary tone source. Twelve oscillators are tuned to the frequencies which correspond to the notes of the top octave of the instrument. Note frequencies for the lower octaves are generated by cascaded frequency halving networks.

Another major difference between the Wurlitzer and the Compton and Hammond systems is the method of timbre generation. Instead of combining sunusoidal components to control spectral characteristics (i.e. additive synthesis), the Wurlitzer organ uses subtractive synthesis. Each oscillator output signal is clipped to produce a square wave which contains many harmonics. A selection of filtering circuits is used to provide the spectral formant shaping required for each of the organ stops. Envelopes of both percussive and non-percussive types are implemented using analogue circuitry, and vibrato is incorporated by frequency modulation of the master tone oscillators.

Crowhurst (1975) describes ten organs which are representative of the current state of the art. In so doing, he outlines various approaches to the problems of

frequency generation and timbre colouration (including envelope, tremolo and vibrato effects). Crowhurst also discusses the influence of the available technology on design philosophy, and identifies and comments on three distinct generations of electronic organs - viz., those which use valves and relays, discrete transistors, and medium- and large-scale integrated circuits. It is interesting to observe the variety of techniques used by the ten third-generation organs described: two employ individual oscillators for each note (the Conn and the Rodgers organs), four use the twelve master-tone generator system with cascaded frequency halving circuits to generate the lower octaves (the Baldwin, Gulbransen, Kimball and Yamaha organs), and four use the digital frequency method described in Section 6.5.1 to generate a tone for each note from a single high frequency oscillator using digital dividers (the Allen, Hammond, Lowrey and Wurlitzer organs). It is also worth commenting that the Allen organ uses an entirely digital approach to waveform generation. Each waveshape (i.e. one period of the repetitive signal - see equation 6.1) is constrained so that its second half is the negative reversal of the first half. The first half of each waveshape is stored in a digital memory as sixteen 7-bit samples. The entire waveshape is generated by presenting to a DAC the successive stored samples which correspond to the first half of the waveshape, followed by the same samples negated and in reversed order. This method of waveshape generation is remarkably similar to that used in the system described in Chapter 6 (cf. Section 6.3).

5.2.3 Other Electronic Musical Instruments

An approach widely used - especially in the popular music field - is the extensive modification of sounds produced by conventional musical instruments. This is achieved using amplifiers, filters and modulators to manipulate the spectral characteristics of signals generated by special transducers which are mounted on the instrument.

An early application of this concept resulted in the electronic piano. The soundboard of a conventional piano is removed, so that oscillations in a vibrating string are essentially inaudible. However, since little energy is transmitted as sound, the vibrations are sustained for much longer. A transducer is mounted adjacent to each string to produce a signal which is analogous to the string's vibration. This signal is subsequently amplified and used to drive a loudspeaker. Some control over timbre is achieved by altering the position of the transducer along the string, since several modes of vibration exist. The electronic piano has become obsolete and is no longer in commercial production (Backus, 1970).

Another type of electronic piano (still being produced commercially) is based on a similar principle but uses vibrating metal reeds instead of strings. The reeds are struck by hammers which are operated by the conventional piano mechanism. The sounds produced in this way are quite different to those of a conventional piano because the harmonics generated by a vibrating reed are different from those produced by a string. Vibrato is also incorporated electronically (Backus, 1970).

The well-known electric guitar uses the same principle as the electronic string piano. The vibrations of the strings are monitored by a transducer mounted either on the bridge or body of the guitar. Recently, similar transducers have been developed for violins, cellos, clarinets etc. (cf. Section 9.3). An important feature of these "electrified" instruments is the inherent possibility of control and modification of the resulting sounds (cf. Taylor, 1965; Backus, 1970). Thus, the spectral characteristics can be altered by filtering, or the signal can be used to control other electronic sound sources such as a voltage-controlled synthesiser (cf. Section 5.2.5) or a ring modulator. For example, a ring modulator (cf. Howe, 1975) can be used to produce a rapidly decaying signal whose frequency is half that of the input signal. In this way, an instrumentalist (or singer) can accompany himself with a pizzicato sound pitched an octave lower (Backus, 1970).

The use of signals to control an electronic sound source has recently been coupled with the concept of bio-feedback (Basmajian, 1967). Thus, signals of biological origin such as the ECG (which monitors heart activity), EEG (produced by neuronal activity of the brain) and EMG (produced by muscle activity) are used to control various sound sources (Rosenboom, 1976). The sound sources provide feedback which permits the person being monitored to learn to control in some prescribed manner the characteristics of the appropriate biological activity. For example, bursts of alpha signal (the 8 Hz to 13 Hz portion of an EEG) can be used to turn a tone on and off. Similarly, the magnitude of the short-time average alpha wave can be used to control the

centre frequency of a band-pass filter, and hence to control sound timbres using subtractive synthesis. Quite elaborate systems involving ten or more people, a large voltage-controlled synthesiser and numerous special-purpose control circuits have been devised to permit the performance of "bio-feedback music" (Rosenboom, 1976).

5.2.4 The Tape Recorder

The introduction of the analogue tape recorder in the late 1940's stimulated the development of "electronic music" as it is known today (Davies, 1974). Initially, the approach mostly used was to record sounds such as those of musical instruments, singing, speech, street noises, railway yards, boiling pots, etc. These recordings are then modified, edited and combined to produce complex sequences of sound (as, for example, in the works of P. Boulez, P. Henry, O. Messiaen, P. Schaeffer and K. Stockhausen during the period 1948 to 1953). This branch of music composition was originally called "musique concrète", although this term is no longer applicable since signals of electronic as well as acoustic origin are used.

The tape recorder is still the most widely used medium for electronic music creation, despite its serious limitations in "live" performance. Douglas (1973) points out two reasons for this: the tape recorder is very versatile, and it is relatively inexpensive and widely accessible. Signals recorded on magnetic tape can be monitored, played back, reversed, edited, and altered in speed to produce changes in pitch. Reverberation can be introduced by re-recording sounds played back in a

reverberant room, by superimposing sounds delayed by acoustic or mechanical transmission lines, or by using a tape recorder with closely spaced multiple recording heads. Complicated rhythmic patterns can be constructed using tape loops. Other signals can be superimposed, and complex sound patterns can be assembled by splicing together edited fragments of tapes.

Tape recorder techniques for electronic music are discussed comprehensively by Douglas (1957), Badings and de Bruyn (1957), Ussachevsky (1958), Beauchamp (1974) and Howe (1975). Voss (1965) and Hiller (1965) describe studios and equipment which are typical of those used to produce many of the innovative compositions of the 1960's (see for example Davies, 1974, for a comprehensive survey).

5.2.5 The Synthesiser

The electronic music synthesiser (also called the formal electronic synthesiser - cf. Olson, 1971) is an integrated system designed specifically for the electronic production and control of sounds. Historically, the synthesiser evolved from the individual items of signal generating and processing equipment used in the early tape studios. Initially, these items (oscillators, filters etc.) were connected together using patch cords for the audio signals. Control was performed manually, usually via potentiometers and switches.

In 1954 the RCA Synthesiser was built by H.F. Olson and H. Belar (cf. Olson and Belar, 1955). This machine contains a comprehensive set of signal generators, modulators, filters and attenuators which are interconnected

using one of eight patch panels. A facility for portamento (frequency glide) is also incorporated. However, the most important feature of the RCA machine is the provision of programmed control using punched paper tape. Fifteen inch wide paper tape is used, moving at nominally four inches per second. The tape is organised in ten columns, each four "holes" wide, so that a total of 40 bits is provided for each "sampling instant" (of which there are nominally 16 per second). Five columns (i.e. 20 bits) are allocated to each of two independent "instruments". Two tape readers are available, so that up to four independent "instruments" can be sounded simultaneously.

The information required to specify the parameters of each "instrument" is coded as follows. The first two columns (4 bits each) specify pitch frequency - column one defines the normalised pitch frequency, while column two indicates which octave is required. The third column defines the envelope characteristic (attack, steady-state and decay). The waveshape spectrum which is pre-patched on one of eight patch panels is selected by column four, and column five defines the overall sound intensity. A set of manually operated switches which parallels the paper tape reader is also provided, so that sounds can be manipulated interactively without the need for pre-programming a paper tape.

A multi-track magnetic tape recorder is incorporated to permit the recording of sound sequences produced by an edited paper tape. The tape speed is synchronised to the paper tape reader, so that the results of several recording

sessions can be combined. This overcomes the limitations imposed by the provision of only four independent "instruments".

Despite its practical disadvantages (high cost, large size and the need for frequent maintenance), much interest was generated by the RCA machine. Several significant compositions were formulated using it, notably Babbitt's (1961, 1964) "Composition for Synthi" and "Philomel", and Wuorinen's (1970) "Times Encomium".

Another programmable synthesiser is the Oramic system (Douglas, 1973). Unlike the RCA synthesiser which uses binary coded input, the Oramic synthesiser uses direct graphical input. A number of transparent perforated film strips are arranged to pass in parallel over a platform when driven by a sprocket. Functions of time are drawn on the film strips using opaque adhesive tape or felt pen. When the sprocket shaft is rotated, all strips move simultaneously over photocells which are constantly illuminated. The pattern on each strip modulates the light, causing a corresponding analogue voltage in the photocell output (cf. the digital graphical encoding device described in Section 6.7). The various signals produced in this way are used to control pitch, duration, vibrato, envelope and intensity, and to select one of a number of waveshape generators. A waveshape is produced by placing a suitably shaped mask over the screen of an oscilloscope which is coupled to a photomultiplier and an oscillator (the pitch source). This causes the shape of the mask to be superimposed on the oscillator output, thereby producing a

periodic signal whose repetition frequency is governed by the oscillator frequency. Changes in waveshape are affected by changing the mask.

The development of the solid-state voltage controlled oscillator in the early 1960's led directly to a new generation of synthesisers. These machines are characterised by the use of electrical rather than mechanical analogue control signals, so that rapid, accurate changes can be made to oscillator frequencies, filter characteristics or amplifier gains. The commercially available synthesisers of Moog (1965, 1967; see also Chadabe, 1967), Buchla Associates, ARP Instruments, Inc. and Electronic Music Studios (London), Ltd, are well-known examples, while numerous systems have been built privately for specific studios (cf. Kindlmann and Fuge, 1968; Scott, 1975).

Typically, voltage controlled synthesisers contain several oscillators whose waveshape (sinusoid, sawtooth, square, pulse) is controlled by switches. Usually one or more noise generator is also included. The portable VCS-3 synthesiser which is manufactured by E.M.S. (London) Ltd. has three independent oscillators and one noise generator, while the E.M.S. Synthi 100 (which is a large machine intended for studio work) has nine independent oscillators and three noise generators. The signals produced by these generators are patched into various filters, spectrum shapers (i.e. contiguous band-pass filters acting in parallel), modulators (amplitude, frequency, phase or ring), reverberators etc., which control and shape the resulting

spectrum using either subtractive or modulation synthesis. Next, envelope effects are introduced by multiplying the synthesised signal by the output of an envelope generator, using a voltage-controlled amplifier. Finally, various sounds which are synthesised in parallel are combined, amplified and fed into a loudspeaker and/or tape recorder.

The most common method of interconnecting the various components outlined above is to use a patch panel and patch cords (similar to those used on analogue computers). However, many instruments now use a matrix panel in which any module output (indexed by row) can be connected to any module input (indexed by column) by inserting a special pin in the appropriate position. It is generally accepted that the matrix patch panel is more convenient to use than patch cords, and eliminates many of the cross-talk problems encountered with the latter (cf. Douglas, 1973). Nevertheless, the matrix panel does have its limitations, since with some designs multiple inputs or outputs are not possible (cf. Howe, 1972) - a criticism more applicable to composition than to performance.

A recent development of the matrix panel is the use of prewired patches, such as the "prestopatch" used by the Synthi AKS, or the "programboard" used by the Buchla Music Easel (Howe, 1975). It is worth noting that the use of a digitally-controlled patch panel has also been considered (Kindlmann and Fuge, 1968; Howe, 1975; see also Section 6.1). This would permit timbre changes to be effected in real-time. However, this facility has not appeared, presumably because of the cost involved (cf. Section 6.1).

Various methods are used to control in real time the performance of pre-patched sounds. The most common of these is the keyboard, which may be monophonic (one note at a time) or polyphonic (two or more simultaneous notes). Howe (1972) discusses some of the operational limitations of keyboards - in particular their unsuitability for portamento or scales other than the conventional 12 tone scale, and the difficulty of controlling several distinct timbres simultaneously from one keyboard. The latter problem is discussed further by Hovey and Seamans (1975) who propose several solutions. The former problem is overcome using a portamento bar, or by using a programmable keyboard whose pitch values can be assigned by the user (Smoliar, 1972; see also the EMS Synthi 100).

A paper tape control system (the "coordinome") which is similar to that used by the RCA synthesiser has been developed for the voltage-controlled synthesiser (Ghent, 1967a; 1967b; Moog, 1967). However the subsequent development of control devices which use electronic memory (e.g. shift registers) have rendered the coordinome obsolete. Caine and Ciamaga (1967) describe the design philosophy of the Serial Sound Structure Generator, which permits the simultaneous and continuous reading of four series containing from 4 to 13 terms each. When each series has been read an automatic wrap-around feature is invoked, so that each series is cyclic. Real time editing facilities are also provided. A similar concept is used by the "sequencer" which was first incorporated in the Buchla synthesisers (Beauchamp, 1974). The sequencer used on the E.M.S. Synthi

100 provides storage and editing facilities for up to 256 events, each with six parameters. Digital memory is used throughout, with one 36 bit word defining the six parameters of each event. Sequences are recorded using a special touch-sensitive keyboard - the start time, duration, key position (which does not necessarily correspond to pitch) and key velocity are recorded each time a key is depressed. The keyboard sampling rate is governed by a clock whose frequency is manually controllable. Successive "layers" of parameters can be recorded, and individual events can be edited (cf. performance editing with foot-switch, Section 3.8). Recorded sequences can be played back at any speed over a range of ±1000 to 1, so that time reversal is possible. A comprehensive description of the sequencer is given by Douglas (1973).

The use of a digital computer permits a more general approach to the control of analogue synthesisers (cf. Zinovieff, 1968; Mathews and Moore, 1970a, 1970b). This approach is well illustrated by "GROOVE" (Mathews and Moore, 1970a, 1970b), which uses a computer to monitor the actions of the human performer (playing a keyboard, twiddling knobs, etc.). These control actions are stored as digitally sampled functions of time, which may be subsequently edited and "played back" through DACs to produce the appropriate control voltages. However, the most useful application of computer control is in what Mathews, Moore and Risset (1974) call the "conductor concept". Thus the actions of the human performer are monitored and used to provide in real time nuances of performance such as changes in tempo, selection

of different timbres, and changes in the relative loudness of different "instruments". This approach (whether used in a hybrid system such as GROOVE or in a totally digital system) offers the greatest potential for real-time interactive music generation. Digital computer techniques are discussed further in the following section.

5.3 DIGITAL SYSTEMS

The analogue synthesis techniques discussed above permit a wide range of sounds to be produced and controlled. Nevertheless, none of the systems described so far can be regarded as universal. For example, those synthesisers which employ subtractive synthesis suffer from the limitation that any transient imposed on the sound spectrum affects all harmonics identically. Additionally, analogue systems are subject to problems of drift, stability, repeatability and cumulative degradation which can be frustrating to a serious composer. In contrast, digital techniques are free of these limitations. This applies whether the implementation is in special-purpose hardware, or in software on a general-purpose computer. Since any conceivable arbitrary real signal can be generated digitally, the potential exists for the development of a universal sound synthesiser.

There are three basic techniques for producing sounds with a computer. These are: sign-bit extraction, digital to analogue conversion, and computer control of special-purpose sound generating hardware. Sign bit extraction is used to produce square waves at controllable frequencies.

It is an inexpensive but limited technique since there is no provision for detailed waveshape or envelope control. Some musical works have been produced using this method - for example "Computer Cantata" by Hiller and Baker (1964) and "Sonoriferous Loops" by Brün (1964).

Digital to analogue conversion is a direct digital synthesis technique in which successive signal samples are calculated or retrieved from memory and presented to a DAC (Mathews, 1961, 1963, 1966, 1969; Tenney, 1963). Real-time operation using a general-purpose computer is, in general, possible only using pre-calculated stored samples (for example cyclic table look-up, cf. Section 2.1). Most systems which use direct digital synthesis require the signal samples to be calculated and stored on digital magnetic tape or disc for subsequent sound generation. Consequently, these systems are essentially batch-processing rather than interactive, and this is one of the main criticisms of the technique (cf. Beauchamp, 1974, who comments: "Perhaps the greatest challenge before electronics technologists in the field of music is to develop new systems that successfully combine the flexibility and universality of the digital computer with the immediacy of the analog systems").

It is the author's opinion that the third technique - computer control of special-purpose sound generating equipment - is the most promising approach for a real-time interactive system. While special-purpose sound generators include conventional acoustic and electro-acoustic instruments (cf. Sections 2.2 and 3.3) and analogue

synthesisers (cf. GROOVE, Section 5.2.5), the main emphasis here is on digital synthesisers.

Numerous comprehensive systems have been developed for "pure" direct digital synthesis (i.e. systems which use a general-purpose computer plus DACs). The best known of these is MUSIC V (Mathews, 1969) and its predecessors (Mathews, 1961, 1963; Tenney, 1963; Roberts, 1966). Versions have been written both in high-level languages (cf. MUSIGOL in Extended ALGOL; MUSIC4F and MUSIC4BF in FORTRAN IV - see Howe, 1975) and in machine-dependent languages (cf. MUSIC7 for the XDS Sigma-7, ORPHEUS for the CDC-3600 and MUSIC360 for the IBM System 360 - see Howe, 1975). A comprehensive description of MUSIC4BF is given by Howe (1975), while MUSIC V is documented by Mathews (1969).

MUSIC V and its relatives are compilers which operate in several passes on "score" and "instrument" specifications which are usually in the form of punched cards. MUSIC V is organised as follows. In Pass 1 the data cards which specify the score and define the instrument characteristics are read and storage areas are assigned. The principal task of Pass 2 is to sort the score description into chronological time order. A variable metronome function can also be invoked to distort the time scale, thus producing gradual accelerandos and ritardandos or abrupt changes in tempo. The actual acoustic samples are calculated and written on to digital magnetic tape in Pass 3.

The concept of the "unit generator" is used to specify each "instrument". The unit generators "perform functions that experience has shown to be useful". They may

be one of a number of standard building blocks, or alternatively may be specified by the user. Examples of standard unit generators are oscillators, noise generators, envelope generators, filters, adders and multipliers. Unit generators are interconnected in the manner described by each "instrument specification". The resulting "instrument" is essentially a digital simulation of a voltage-controlled analogue synthesiser.

Stereophonic as well as monophonic sounds can be produced. A catalogue of sounds together with a description of the instrument(s) which generated them is available to help composers use the system (Risset, 1970).

Slawson (1969) describes a synthesis system which is similar to MUSIC V in many respects. However, the score and timbre specification language used relates directly to the note sequences and the sounds it represents (see also Dallin, 1974). An interesting feature of this approach is that the basis for the description of timbres is the phonetic sounds of speech. This permits signals whose spectra vary dynamically to be easily notated. The technique used to compute the corresponding signal samples is borrowed directly from speech synthesis. A "terminal analog" model of speech production is used, in which the acoustic energy source is assumed to be independent of the vocal tract (cf. Section 7.3). A four pole transfer function with no zeros is used to model the vocal tract. Each phoneme is assigned a mnemonic with which is associated a table which specifies the corresponding time-varying vocal tract transfer function and the mode of excitation (i.e. voiced or

unvoiced). The phoneme description tables are embedded within the synthesis program and are essentially transparent to the composer. Several noteworthy compositions have been produced using this system, in particular Slawson's (1967) "Wishful Thinking About Winter" and (1968) "Movements for Orchestra with Tape".

Recognition of the importance of immediate feedback in the composition process has motivated the development of several systems which achieve real-time operation. The programs POD4, POD5 and POD6 (Truax, 1973a, 1973b) use "pure" digital synthesis and are designed specifically for a medium sized computer (a PDP-15 with 12K 18-bit words core memory). Up to five 12-bit DACs are used. The main design philosophy is to provide real-time interaction for composers rather than to implement a comprehensive sound synthesiser. For this reason the range of sounds which can be produced is comparatively limited. POD4 and its successor POD5 use fixed waveform synthesis, in which one period of the required waveshape is stored as 50 equally-spaced samples and the sound is generated using cyclic table look-up. To accommodate pitch frequencies up to 5150 Hz (approximately E8 - see Table 9.2) the number of samples per period is reduced to 25 and 17, depending on the pitch. Envelope and vibrato (amplitude modulation) effects are implemented by storing ten different versions of each waveshape. Each version corresponds to the basic waveshape scaled by a suitable amplitude factor, so that in effect an envelope or vibrato is limited to ten discrete amplitude steps. Waveshapes can be retrieved from a "sound catalogue"

which is stored on disc or magnetic tape, calculated using harmonic (Fourier) synthesis, or read from paper tape (cf. Section 6.7). POD6 uses frequency-modulation synthesis (see Section 5.4), although the method used to achieve real-time operation is not clearly described.

Smoliar (1973) describes a comprehensive data structure called EUTERPE2 which is designed for real-time interactive composition using up to six independent "voices". However, Smoliar does not discuss the method of sound generation used, and it appears likely that each "voice" is realised using special purpose hardware.

Groove (Mathews and Moore, 1970a, 1970b) has already been mentioned (Section 5.2.5) as an example of a "hybrid" synthesis system in which a digital computer is used to control an analogue synthesiser. MUSYS (Grogono, 1973) is similar in that it uses digitally-controlled analogue hardware (a bank of oscillators, noise generators, a percussion simulator, envelope shapers, filters and reverberators). Digitally-controlled function generators and several DACs are also incorporated. These components are manually patched using four (32 × 32 element) matrix panels, although a facility is provided to permit limited digitally-controlled matrix switching. MUSYS is organised in four distinct modules:

(i) A Text Editor, which the composer uses to prepare or modify score encoding data (written in the MUSYS language)

(ii) A Compiler, which converts the score into coded data lists

(iii) A Performance Program, which reads the compiled data from disc and prepares it for delivery

(iv) A Delivery Program, which sends data to the appropriate devices in real time.

Grogono comments that the speed of the overall system is "adequate for interactive use", and that the procedure usually adopted by composers is to divide a score into small units which can be compiled in a few seconds. The main criticism of the system (and this applies to MUSIC V and its variants also) is that the composer must learn a special language with which to encode a score. As Grogono (1973) comments: "It is tempting to design a new language which makes more use of keywords and is not so barbarously hieroglyphic as MUSYS".

Taylor (1972) sets out an eloquent argument in favour of a "hybrid" synthesis system in which special purpose digital rather than analogue hardware is used in the "black box" sound generating units. Taylor details comprehensive proposals for such a system, which includes numerous analogue input devices, keyboards, and a graphics system to aid composer interaction. The digital hardware synthesiser proposed contains 128 oscillators, 128 multi-purpose filters, 128 envelope generators, 4 noise generators, 4 reverberation units, several high-speed arithmetic units capable of real-time amplitude, frequency and ring modulation, and facilities for controlling sound movement in two or three dimensions (cf. Section 5.5). This hardware complement permits timbres to be produced using both additive and subtractive synthesis as well as direct

waveform synthesis and modulation synthesis. At the organisational level, Taylor proposes the development of simulation syntax compilers to permit the performance of compositions already encoded in languages such as MUSIC V. Incorporation of the "Music Box" language concept developed by K. Wiggen (cf. Sandlund, 1972) is also proposed.

Alonso, Appleton and Jones (1975) describe an all-digital system developed at Dartmouth College. This synthesiser has sixteen independent "voices" and can be used in a time-shared mode by up to four terminals. Thus four users can simultaneously produce four-part musical passages. Both the direct waveshape synthesis and frequency modulation synthesis methods are used. A similar system which uses additive, subtractive and frequency modulation synthesis is described by Gross and Leibig (1976). The latter system is oriented towards instruments whose timbres are time-varying during each note (cf. Section 5.4).

The digital hardware synthesisers discussed above are all implemented using dedicated hardware. A more flexible approach is the programmable digital processor, which is a special purpose computer designed specifically for high speed signal processing (see also Section 8.5). Blesser, Baeder and Zaorski (1975) describe a synthesis-oriented machine whose architecture incorporates partial pipelining, three arithmetic data registers and a high speed (less than 50 nanosec. access time) scratch memory in addition to main memory. Each executable instruction (e.g. load accumulator, add, multiply, compare, skip on condition, etc.) can be combined with any other instruction

to form a "compound instruction". Both component instructions are executed in parallel. In this way, the effective instruction cycle time (200 nanosec.) is halved. Thus for a 35 kHz signal sampling rate, up to 285 component instructions can be executed for each sample while achieving real-time operation.

It is worth commenting that while high-speed programmable processors provide great flexibility and convenience for experimentation, design and evaluation of synthesis techniques, they still operate serially rather than in parallel. However, the complex sounds of interest in music are intrinsically parallel (i.e. combinations of different "instruments"). This parallelism is explicitly exploited in most composition-oriented synthesisers. While a serial machine can simulate parallel operations, it is likely that the speed of operation of most programmable serial processors available today is still too slow to satisfy the real-time synthesis requirements of serious composers.

5.4 DIGITALLY ORIENTED TECHNIQUES

The systems discussed in the preceding sections use one or more of four techniques for timbre synthesis - viz. additive synthesis, subtractive synthesis, direct waveform synthesis and modulation synthesis. In this section the characteristics and limitations of these methods are briefly considered in relation to the timbral characteristics of conventional acoustic instruments. Several other methods which are suitable for digital implementation are also

discussed. These latter synthesis techniques offer advantages in hardware realisation or in the ease with which time-varying timbres can be controlled or produced.

The unnatural character of many synthesised sounds of "pure" electrical origin is well known. This unnaturalness is due to lack of variation in the waveshape within each note. In contrast, the sounds produced by acoustic instruments are characterised by inter-period changes in the waveform as well as by a temporal envelope variation (see Section 7.3). This is particularly important during the attack transient (cf. Robson, 1976). Thus the ideal synthesis system should permit independent control of the envelope of each of the spectral components of a sound. The frequency and relative phase of each spectral component should also be controllable, so that enharmonic as well as harmonic components can be used.

The limitations of direct waveform synthesis and subtractive synthesis are obvious and are discussed respectively by Xenakis (1971) and Howe (1972). Additive synthesis permits the production of a much wider class of sounds (cf. Beauchamp, 1974). Its implementation using analogue hardware suffers from the disadvantage that a large number of oscillators and envelope generators are required (cf. Howe, 1972). Also, the relative phases of the spectral components are not easily controllable (cf. Crowhurst, 1975, who comments on the desirability of randomising the phases of the spectral components of synthetic organ sounds at the start of each note). Neither of these objections is applicable if the FFT is used. Ironically, the main

disadvantage of additive synthesis for performance (as distinct from composition) seems to be the number of control parameters required to specify each sound (cf. Howe, 1972).

The Walsh transform (see Section 8.4.1) has also been applied to additive synthesis (Hutchins, 1973, 1975; Insam, 1974). This approach requires that the time-base of the Walsh transform be chosen as the pitch period, and that the Walsh spectrum rather than the Fourier spectrum be used to specify the required sound. Unfortunately the Walsh transform is not invariant to phase changes of the corresponding Fourier spectral components, so that a simple analogy between the Walsh and Fourier spectra does not exist.

Modulation synthesis, while not as general as additive synthesis, does permit the production of a large variety of time-varying timbres using comparatively few control parameters. Frequency modulation in particular has recently received much attention (cf. Chowning, 1973; Alonso, Appleton and Jones, 1975; Scott, 1975; Moorer, 1976). The main advantages of this approach are the ease with which complex spectra are synthesised, and the simplicity with which the temporal evolution of the frequency components is controlled. This latter feature is achieved at the expense of lack of independent control over the spectral components, so that frequency modulation is not useful for modelling realistically the sounds of orchestral instruments (cf. Chowning, 1973). Nevertheless the technique can be used to synthesise sounds which possess characteristics similar to orchestral instruments, as Chowning demonstrates.

To illustrate frequency modulation synthesis,

consider the case where both the carrier and modulating signals are sinusoids. Constrain both signals to lie in the audio band, so that the resulting sidebands form the synthesised spectrum directly. Then

$$s(t) = A \sin (\alpha t + I \sin \beta t) \qquad (5.1)$$

where $s(t)$ is the modulated signal, $\alpha = 2\pi f_c$ is the carrier frequency (rads/sec), $\beta = 2\pi f_m$ is the modulating frequency (rads/sec), $I = d/f_m$ is the modulation index, and d is the peak deviation (see for example Lathi, 1965). The bandwidth (and hence the spectrum complexity) depends on both the frequency deviation and the modulating frequency according to:

$$B \simeq 2(d + f_m) \qquad (5.2)$$

where B denotes bandwidth (cf. Chowning, 1973). Thus, if the modulation index I is time varying, the evolution of the bandwidth of the resulting spectrum is characterised by the form of $I(t)$. However, the evolution of each spectral component also depends upon $I(t)$, since as I increases energy is transferred between the carrier and an increasing number of side frequencies. This dynamic interdependence between modulation index, bandwidth and spectral amplitude is illustrated graphically by Chowning (1973).

Moorer (1976) extends Chowning's method in a form suitable for digital implementation. Moorer's starting point is the discrete summation formula (cf. Jolley, 1961)

$$\sum_{k=0}^{N} a^k \sin(\theta + k\beta) =$$

$$\frac{\sin\theta - a\sin(\theta-\beta) - a^{N+1}[\sin\{\theta + (N+1)\beta\} - a\sin(\theta + N\beta)]}{1 + a^2 - 2a\cos\beta} \qquad (5.3)$$

As the number N of components becomes infinitely large, equation (5.3) simplifies to

$$\sum_{k=0}^{\infty} a^k \sin(\theta + k\beta) = \frac{\sin\theta - a\sin(\theta-\beta)}{1 + a^2 - 2a\cos\beta} \quad , \quad a < 1 \ . \qquad (5.4)$$

These expressions (5.3) and (5.4) are applied to sound synthesis by setting $\theta = 2\pi f_c t$ and $\beta = 2\pi f_m t$, where f_c and f_m are, respectively, the carrier and modulating frequencies in Hz. Both harmonic and enharmonic spectra can thus be produced, depending on whether the ratio f_m/f_c is rational or irrational. The special case $f_c = f_m$ is noteworthy since it permits further simplification of the right-hand side of equation (5.4). Observe however that whenever equation (5.4) is used the ratio a of adjacent spectral components must be chosen small enough that the spectrum is effectively bandlimited to half the sampling frequency, to avoid aliasing.

Equations (5.3) and (5.4) express "one-sided" spectra, which are asymmetrical about f_c. To produce "two-sided" spectra which are symmetrical about f_c, the following extension of (5.3) can be used:

$$\sin\theta + \sum_{k=1}^{N} a^k \sin(\theta + k\beta) + \sin(\theta - k\beta) =$$

$$\frac{\sin\theta(1 - a^2 - 2a^{N+1}[\cos\{(N+1)\beta\} - a\cos N\beta])}{1 + a^2 - 2a\cos\beta} \qquad (5.5)$$

As N becomes infinitely large this simplifies to:

$$\sin\theta + \sum_{k=1}^{\infty} a^k\{\sin(\theta + k\beta) + \sin(\theta - k\beta)\} =$$

$$\frac{(1 - a^2)\sin\theta}{1 + a^2 - 2a\cos\beta} \quad , \quad a < 1 \quad . \qquad (5.6)$$

The application of the discrete summation formulae (5.3) to (5.6) to compute signals whose spectral envelopes vary with time is discussed by Moorer (1976), who presents several examples. The main advantage of this approach over Chowning's (1973) frequency modulation technique is that greater control can be exercised over the form of the spectrum (within the constraints imposed by the left-hand side of equations (5.3) to (5.6)). However, the discrete summation formulae are comparatively complicated to compute. Another disadvantage is the dependence of the overall signal amplitude on the factor a, which is an important control parameter for the generation of dynamically-varying spectra. This objection can be overcome by using "denormalisation factors" which maintain constant signal power as a varies.

An approach which is quite different to the deterministic synthesis techniques discussed above is suggested by Xenakis (1971), who proposes that sounds be generated stochastically. Xenakis envisages "the pressure variations produced by a particle capriciously moving around equilibrium positions along the pressure ordinate in a non-deterministic way". Various probability distribution functions are suggested by Xenakis (e.g. Poisson, exponential, normal, uniform, Cauchy, Bernoulli, logistic),

both individually and in combinations. He also considers the use of randomised or deterministically-varying parameters (e.g. time) and constraints (e.g. elastic or inelastic barriers) in the probability distribution functions. Xenakis presents graphical examples of various signals generated in this manner. Unfortunately it is not possible to judge the aural effect of these signals without hearing them.

Linear prediction (cf. Section 7.6) is now widely used in speech analysis and synthesis, although to the author's knowledge it has not been used for the synthesis of musical sounds. The recursive filter formulation used in linear prediction is outlined in Section 7.6, together with a discussion of the assumed signal model, and is not repeated here. However it is pointed out that the speech model which is explicitly used in the formulation is not directly applicable for musical instruments (cf. Section 7.3). Nevertheless, the recursive filter method does permit the synthesis of signals with dynamically-varying spectra using comparatively few control parameters which relate directly to the intuitively important characteristics of sounds (viz. the excitation, the formant frequencies and bandwidths, the pitch frequency, and the intensity). The usefulness of the method can be judged by listening to the record which accompanies the paper by Atal and Hanauer (1971).

Linear prediction synthesis could be applied directly to a music synthesiser of the kind described by Slawson (1969) (cf. Section 5.3). It is also probable that sounds

with characteristics similar to those of orchestral instruments could be synthesised by direct application of the speech analysis-synthesis technique. However, further research is required to ascertain whether linear prediction synthesis can be successfully applied to a wider class of signals than that assumed by the speech model. In particular, the dependence of pitch on the dominant formant poles, and the use of excitation signals other than the impulse train and white noise should be investigated. The use of deterministic or stochastic second-order variations into the filter coefficients should also be considered as a technique for introducing "realism" into the resulting sounds.

5.5 CONTROL OF SPATIAL SOUND EFFECTS

Spatial effects are becoming an increasingly important feature of electronic music. Fellgett (1973) points out that explicit attention to the spatial content of music is not a new phenomenon, and cites the work of Monteverdi (17th Century), J.S. Bach (18th Century) and Stravinski (20th Century). Conventional multi-channel sound reproduction preserves some of the directional and distance cues of the recorded sound field to produce an illusory acoustical space (see Fellgett, 1973, for a discussion of stereo, quadraphonic and ambisonic sound reproduction). This section briefly considers the incorporation of similar illusory spatial effects in synthesised music.

The factors which influence the perception of the direction and distance of a sound source are not well understood (cf. Bui, 1977). Mathews (1969) observes that transient effects are important, and that the first arrival of a sound is more significant than subsequent (reverberant) arrivals. This latter phenomenon is called the "precedence effect" (cf. Wallach, Newman and Rosenzweig, 1949). The work of Gardner (1962, 1967, 1969) is also relevant here.

The information which is required by a listener to locate an actual sound source in an enclosed space is of two kinds - that which defines the direction of the source relative to the listener, and that which defines the distance of the source from the listener. Chowning (1971) lists the cues for angular and distance location as follows:

(i) The different arrival times of the signal at the two ears when the source is not centred in front of or behind the listener.

(ii) The pressure level difference at the two ears, resulting from the shadow effect of the head when the source is not centred. This effect is more pronounced at high frequencies (cf. Flanagan, 1972).

(iii) The ratio of the direct energy to the indirect (i.e. reverberant) energy. (The former decreases more rapidly with distance than does the latter.)

(iv) The loss of low-intensity frequency components of a source with increasing distance.

Cues (iii) and (iv) are a consequence of the non-linear attenuation of sound with distance.

Chowning (1971) describes a preliminary system

designed to incorporate spatial effects into music synthesised digitally using a program similar to MUSIC V (cf. Section 5.3). Four separate sound channels are used, and spatial control is effected with a display screen and joystick. Angular location cues are provided by adjusting the ratio of the direct signal energies applied to each loudspeaker pair. This approach is used because the precise location of the listener is not known *a priori*, so that those localisation cues which are dependent upon delay, phase, and the orientation of the listener's head are inappropriate. Distance cues are provided by controlling the ratio of the direct signal to the reverberant signal. A Doppler shift effect is also incorporated to provide a velocity cue, so that moving sources can be simulated.

Research into techniques for the creation of illusory sound spaces is continuing. Chowning, Grey, Rush and Moorer (1974) describe this as one of their main research goals, together with the related topic of artificial reverberation. It is apparent that much work is still required on both the perception and simulation of distributed sound fields.

CHAPTER 6

A COMPUTER CONTROLLED DIGITAL SYNTHESIS SYSTEM

6.1 RATIONALE

The digital synthesis system described in this chapter was constructed to overcome limitations inherent in the electronic organ used in the music input/output system described in Chapter 3. The most serious defect of this organ was its use of a separate oscillator for each of its forty-eight notes, resulting in poor frequency stability and the consequent need for frequent tuning. Another disadvantage was its lack of timbre control, since it possessed only two voice stops. While the latter was of little consequence for many system applications, it placed severe restrictions on the potential of our system for composition and performance.

A strong motivating and formulative influence on our work was our regular discussions during 1973 and 1974 with J.E. Cousins of the Music Department. These introduced us to the synthesiser and tape recorder techniques, used by electronic music composers, which are reviewed in Chapter 5. Our primary objective was to implement a performance-oriented system similar to GROOVE (cf. Section 5.2.5), in which the performer's control actions are monitored in real-time and used by the computer to generate switching and control signals for a synthesiser. Thus, we envisaged the

possibility of rapid changes to the synthesiser patching, so that stored patching configurations corresponding to new or existing sounds are retrieved and implemented during real-time performance. While computer-controlled patching is technically feasible and forms the basis of new hybrid computing systems (Rubin, Keener and Downer, 1975), it is not practicable with our limited resources because of the intrinsically parallel nature of the switching matrix and the dimensionality factor encountered with increasing synthesiser complexity. This latter is demonstrated by comparing the rudimentary EMS VCS-3 three oscillator synthesiser, whose 16 × 16 matrix requires 256 analogue connections, with the more sophisticated EMS Synthi AKS used by the Victoria University of Wellington electronic music studio, which has 4,800 cross connections in its control matrix.

The problems of automating a matrix patch panel, together with the intrinsic problems of stability, drift and repeatability associated with analogue voltage controlled systems, led us to conclude that an entirely digital approach to sound synthesis is more appropriate. It is interesting that our conclusions agree with those of other researchers working independently in this area (cf. Scott, 1975; Alonso, Appleton and Jones, 1975).

The considerations discussed above led to the development of the computer controlled digital synthesis system described in the remaining sections of this chapter. The fundamental system design was formulated in late 1974 by W.K. Kennedy and the author, with assistance from

Professor R.H.T. Bates, M.R. Lamb and Susan D. Frykberg. Initial development work was undertaken in 1975 by R.J. Howarth and R.G. Vaughan as an undergraduate project (Howarth, 1975; Vaughan, 1975; Tucker *et al.*, 1975), and continued during the summer vacation under a research assistance grant. It was pursued during 1976 by Vaughan as a Master's project (Vaughan, 1977), with related high-level software development by Susan Frykberg. The author's role throughout has been that of overseer and system co-ordinator, with personal responsibility for the computer interface hardware and software.

6.2 SYSTEM OVERVIEW

The fundamental design objectives of the system, formulated in late 1974 and early 1975, were as follows:

(a) High frequency stability of note pitches corresponding to both the equally-tempered scale and to other user-defined scales or frequencies. This facility permits glissandi (continuous pitch change) effects, and enables the generation of both microtonal and conventional music.

(b) A wide range of voice timbres, together with a flexible interactive method of specifying them, and facilities for rapid timbre changes during a performance. Independent control over the periodic waveshape, envelope, and modulating vibrato and tremolo signals is required.

(c) Two modes of operation, manual or organ mode, and computer-controlled or synthesiser mode. In organ mode the device parallels a conventional electronic organ, and

is played manually from a keyboard. Note timbres can be changed in real-time by "voice stop" switches. In synthesier mode the "note event" sequences (start time, pitch, duration, intensity and timbre) are computer controlled. Both sequencer control (cf. EMS Synthi 100 "DIGI") and programmed control (cf. MUSIC V) are thus possible.

(d) A modular design and implementation capable of future expension. This approach is necessary for several reasons. The available budget for the initial system, which was completed in February 1977, was approximately $1000 N.Z. This figure excludes the computer interface and software development. In addition, a modular system design provides the flexibility necessary in prototype development.

These objectives and constraints led to the design outlined in Figure 6.1. It incorporates digital frequency generation, pulse code modulation techniques, and a high level of time-shared, rather than parallel, hardware. This last factor in particular is a significant departure from conventional synthesis techniques, since it permits highly cost-effective hardware utilisation. Thus, the use of more sophisticated digital, as opposed to analogue, signal generation techniques is justified on economic as well as performance grounds.

The synthesis system consists of a number of independent "voice units", together with a single control unit which handles timbre changes and note playing under computer direction. Each "voice unit" generates a single composite sound composed of periodic waveshape, envelope,

tremolo and vibrato components. It can be regarded as a single unisonous instrument (cf. woodwind and brass instruments), with both percussive and non-percussive characteristics available. Thus, the total maximum number of simultaneous notes which can be played at any instant equals the number of "voice units". Since all voices are independent, both chord effects (cf. keyboard instruments) and ensemble effects (cf. orchestral ensemble) may be produced.

The system modularity permits additional voice units to be added, with an upper limit imposed by the time-shared control unit. We have implemented two voices, and designed for a maximum of sixteen. The expandable modular hardware concept is reflected explicitly in the software design, which may be easily modified to accommodate both additional voices and more control parameters for each voice.

Each voice unit consists of a pair of recirculating digital memories. The waveshape memory contains the coded samples corresponding to one period of a repetitive signal. It is so arranged that sequential signal samples are cyclically presented to a DAC at a rate governed by a digitally controlled clock. Thus, the waveshape is reconstructed as a periodic analogue voltage with a repetition frequency (and hence pitch) dependent on the memory clocking rate (see Section 6.3). The envelope memory operates similarly, but with additional control circuitry to ensure that precisely one complete memory circulation occurs in the interval corresponding to any specified note duration. Additional envelope control permits the implementation of

both percussive and non-percussive envelopes by including an unclocked steady state portion in the latter.

The analogue envelope and periodic waveshape, together with tremolo if desired, are combined using the analogue techniques discussed in Section 6.3. Vibrato is incorporated by frequency modulation of the waveshape memory clock. Individual gain control of the resultant "voice" provides independent control over the relative loudness contributions of simultaneously occurring notes.

At the beginning of a performance or composition session, the user selects the timbre he desires from a sound catalogue stored on disc memory (cf. Mathews, 1969; Mathews, Moore and Risset, 1974) or specifies the waveshapes and envelopes using one of the specification facilities discussed below. The voice load controller transfers the corresponding signal samples from the computer to the selected voice waveshape or envelope memory. Play control of individual voices is effected by the play controller under software direction - thus a specified voice may be initiated at any specified audio frequency, and with either percussive or non-percussive envelope. Once activated, that voice will continue independently of all other voices until it is turned off by a software control signal (non-percussive mode), or until its percussive envelope has decayed. Since control of voice load and play facilities is at the level of individual voices, different timbres may be played simultaneously, and the sound produced by any voice may be altered between notes.

High-level play control software permits the system

to be used in either organ mode - i.e., played from the organ keyboard which is monitored by the computer - or in synthesiser mode from a pre-recorded, edited table of note parameters stored in the computer. At the time of writing only the sequencer option of synthesiser mode is operational. This is an extension of the playback system discussed in Chapter 3, with the inclusion of timbre control by pre-stored parameters governing voice characteristics, or in real time by monitored "voice switches". The development of a high-level music language compiler similar to MUSYS (Grogono, 1973) or MUSIC V (Mathews, 1969) is at present under consideration. (These compilers are discussed in Section 5.3.) The inclusion of a similar facility in our system would permit greater control and flexibility for the composer, and enable the direct performance of existing compositions which are already encoded in these languages.

The remaining sections of this chapter discuss the system in detail.

6.3 SIGNAL GENERATION

The model of sound generation we have adopted requires four independent signal components to synthesise each "voice" signal - a periodic signal we call a "waveshape", an envelope which multiplicatively modifies the waveshape, tremolo which provides periodic amplitude modulation of the envelope, and vibrato which modulates the waveshape periodicity.

Denote the composite signal by $s(t)$, and let $w(t)$, $e(t)$, $\tau(t)$ and $v(t)$ be the waveshape, envelope, tremolo

and vibrato signals respectively. Then

$$s(t) = G\, e(t)\, [1 + \tau(t)]\, w(v(t)) \qquad (6.1)$$

where G is a gain factor.

This model for sound synthesis is not unique, nor is it entirely satisfactory for realistic synthesis of conventional music instruments (Richardson, 1954; Luce, 1963; Strong and Clark , 1967a, 1967b; Freedman, 1967; Keeler, 1972; Robson, 1976; see also Sections 5.4 and 7.3). Despite these limitations it permits the real-time generation of a wide variety of tone colours, and is formulated in a manner which permits the individual contributions of waveshape, envelope, tremolo and vibrato to be investigated (cf. frequency modulation synthesis, which is discussed in Section 5.4).

Waveshape generation is effected using pulse-code modulation as follows. Digitally coded samples corresponding to one period of the waveshape are stored in a recirculating shift register memory. As the memory is clocked, successive samples are presented to a DAC so that the original periodic waveshape is reconstructed as an analogue voltage. Control of the memory recirculation rate, and hence of the pitch of the resulting waveshape, is achieved by altering the frequency of the memory clock according to

$$f_c = N\, f_o \qquad (6.2)$$

where

f_c = frequency of shift register memory clock (6.3)

N = number of words in memory (6.4)

f_o = repetition frequency of output periodic waveshape (6.5)

From the Sampling Theorem (Shannon and Weaver, 1959), it follows that this technique is capable of regenerating any signal band-limited to f_B where

$$f_B = \frac{f_c}{2} \quad . \tag{6.6}$$

Thus N/2 spectral harmonics may be synthesised. In our implementation N is chosen to be 128. This is a compromise between the conflicting requirements of high temporal and spectral resolution, low cost and the desirability of covering the audio range

$$50 \text{ Hz} < f_o < 5 \text{ kHz} \tag{6.7}$$

while maintaining the constraint

$$f_c \leq 1.0 \text{ MHz} \tag{6.8}$$

imposed by the available memory technology.

An eight bit amplitude resolution was adopted as a reasonable compromise between memory cost and quantisation error. This yields a normalised quantisation error of about $\frac{1}{4}$%, with a rms signal/noise ratio given by $2^8[3/2]^{\frac{1}{2}}$ or 50 dB for a normalised sinusoid (Bennet, 1948; Mathews, 1969).

The limited dynamic range available with 8 bit resolution is extended by using normalised (full scale) signals, with separate gain control. An alternative

technique for dynamic range extension which is commonly used in speech PCM systems is digital logarithmic companding, which employs a logarithmic rather than linear encoding characteristic to achieve a dynamic range comparable with 12 bit linear encoding in only 8 bits (Schoeff and Reid, 1976). While commercially produced logarithmic DAC and ADC units are available (e.g. Precision Monolithics COMDAC series), the additional complexity of signal specification software involved with this approach is not warranted.

An interesting problem arises from the use of a signal sampling rate f_c which is dependent on pitch (equation 6.2), rather than constant. This requires the regenerated analogue signal to be low-bass filtered with a cut-off frequency given by (Lathi, 1965)

$$f_L = \frac{f_c}{2}$$

$$= \frac{f_o N}{2} \qquad (6.9)$$

While low pass filters with controllable cut-off frequency can be constructed (Mitra, 1971), we have found that the use of a stochastic DAC (Insam, 1973) incorporating fixed low pass filtering yields an acceptable analogue reproduction over most of the audio range. At high repetition frequencies, f_o approaches the DAC filter cut-off and limits the high frequency end of the output signal spectrum. With low values of F_o, the DAC filter is unable to remove the duplicate spectrum which commences at $f_c/2$, and which is manifested as a high pitched whistle of low intensity. While it is outside the scope of this thesis

to compare stochastic DAC characteristics with those of conventional converters based on zero-order sample and hold techniques, subjective listening tests and visual waveform inspection indicate that a smoother, less stepped waveform results from the former. This is qualitatively explained by the fact that the stochastic DAC output is essentially white noise with a short-time mean value proportional to the digital input (Insam, 1973; Howarth, 1975). Another alternative to the conventional zero-order hold is higher order interpolation (Cadzow and Martens, 1970; Oetken, Parks and Schüssler, 1975). Thus, a linear or quadratic interpolation smooths the output time function and attenuates the unwanted duplicate spectrum. We consider that the best solution is provided by incorporating a switchable range of DAC filter cut-off frequencies, controlled by the voice pitch.

A similar digital synthesis technique is used for the generation of envelope signals, with the important distinction that the envelope memory clock rate is essentially constant rather than pitch dependent. Also, since the envelope signal is not in general periodic, control hardware is used to ensure that a complete envelope memory circulation occurs in precisely the desired note duration. With the percussive envelope option, the envelope signal is generated continuously at constant rate until completed - i.e. until the envelope "decay" is complete. When the non-percussive option is specified, the envelope is generated in three stages - attack (clocked), steady state (unclocked) and decay (clocked). These two modes of

envelope generation are illustrated in Figure 6.2.

To permit standardisation of waveshape and envelope memory and memory-load hardware (see Section 6.4), it was decided early in the design to use identical components for both waveshape and envelope signal storage. Thus 128 eight bit words are used to define both waveshape and envelope signals. Static rather than dynamic MOS shift register memory was chosen, despite its lower maximum clocking rate. This choice simplifies the memory clocking and output gating, since dynamic memory requires continual refreshing.

Although we desire an entirely digital approach to sound synthesis, the expense of fast digital multipliers for combining waveshape, envelope and tremolo makes analogue signal mixing a more realistic alternative. In addition, since tremolo is essentially a low frequency (0 - 20 Hz repetition frequency) signal and is generally small compared with the envelope and waveshape magnitudes (cf. Section 7.3), it can be regarded as a second order effect in the composite signal. This consideration has led to an analogue implementation of tremolo at this stage of system development.

Vibrato is a second order effect similar to tremolo, but since it is manifested as a frequency modulation of the waveshape memory clock, it is discussed separately in Section 6.5.

6.4 LOAD CONTROLLER

The load controller hardware module is responsible for two tasks which together comprise the voice memory load

operation. Firstly, the serial string of 128 eight bit numbers which are required to specify an envelope or waveshape must be routed from the computer interface to the correct voice, and thence to either the envelope or waveshape memory. This task is essentially a multiplexing operation, and is performed using a cascaded 1 - 2 MUX which selects a waveshape or envelope route, and two 1 - 16 MUX's which select the destination voice. Secondly, the memory control signals RECIRCULATE CONTROL and CLOCK are required to ensure that the memory is in the load rather than store mode, and to synchronise the data storage.

It was found convenient to generate and transmit these memory control signals with the data, so that the multiplexed route is 10 bits wide. The unaddressed state of the RECIRCULATE CONTROL bit MUX is adjusted to coincide with store mode, so that all memories are automatically set to store mode unless a data transfer is occurring.

To provide additional flexibility, provision is made for transfer of data from either the computer interface or directly from the hardware specifier unit which is described in Section 6.7. This is implemented by inclusion of a 2 - 1 MUX at the beginning of the multiplexed path.

6.5 PLAY CONTROLLER

The play control system is responsible for generating the controllable-frequency clock pulses for each voice unit waveshape memory, and for initiating envelope control signals.

6.5.1 Waveshape Control

Because of the piecemeal nature of the system development and the need to obtain results at the undergraduate project level, two modes of frequency control were proposed during the initial design. The first mode, which is now operational, uses a global approach to frequency generation. In this approach, all the 61 frequencies corresponding to the five octave equally-tempered scale C2 to C7 are generated in parallel in the form $f_c = 128\ f_o$ Hz (cf. equation 6.2).

Digital frequency generation is used, incorporating a commercially-available master tone generator (General Instrument Corporation Microelectronics AY-1-0212) driven by a crystal oscillator. The master tone generator uses the pre-programmed frequency division factors listed in Table 6.1 to produce the twelve frequencies corresponding to the octave C#6 - C7, with a worst case error of 0.1%. The tone generators available to us were not capable of producing directly the higher $f_c = 128\ f_o$ frequencies. This problem was overcome by using phase-locked loop frequency multiplication on each of the twelve master tone generator outputs. The five multiplied frequencies corresponding to the lower octaves are then generated using cascaded division by two.

A computer-controlled multiplexer with 64 inputs and up to 16 simultaneous outputs is used to route the pulse train of specified frequency to a selected voice, without altering the switching paths existing to other voices. The MUX is in effect a small computer-controlled electronic

switching exchange capable of handling wide-band digital signals, and is constructed using standard TTL technology. This global approach to frequency generation permits economies in the control interface, since only 6 bits of pitch address, 4 bits of voice address and a MUX enable bit are required to control a 5 octave, 16 voice system. The 64 pitch inputs are allocated so that 61 inputs correspond to the notes C2 to C7, with the remaining 3 inputs held at ground potential. This permits silence (or absence of clock) to be specified in the same manner as a note, by addressing one of the 3 "silence" inputs. This convention is utilised in the hardware which generates the envelope control signals. Thus, the pitch address is decoded to determine whether a note is commencing or finishing, so that an additional control bit is not required.

The second mode of frequency generation uses a local frequency generation approach, in which a wide-band digitally-controlled oscillator is incorporated with each voice. The objective here is to permit the generation of a pitch continuum. It was originally envisaged that this facility be implemented using voltage-controlled oscillators controlled by the hybrid computer DAM facility. However, the higher repeatability and precision afforded by direct digital frequency generation has led to a revision of this idea, and design of a wideband digital oscillator using programmable frequency division from a master crystal oscillator is now being considered. An interesting feature of current proposals (Vaughan, 1977) is the inclusion of hardware vibrato generation. Thus, the control software

specifies the frequency division factor M which corresponds to the desired frequency - this can be directly calculated using

$$f_m = Mf_c \tag{6.10}$$

where

f_m = master oscillator input frequency (6.11)

f_c = output clock frequency (6.12)

M = digital oscillator division factor (6.13)

In addition, the periodic vibrato parameters δM and δT are specified. In this manner f_c can be altered by $\pm \delta f_c$ in the time interval δT, where

$$f_c \pm \delta f_c = f_m / (M \pm \delta M) \tag{6.14}$$

The manner in which fine temporal variations occur within δM must also be specified - this can be achieved using the digital method implemented for waveshape generation, or using pre-selected analogue function generation techniques and an ADC. Thus sinusoidal, triangular or rectangular temporal variations in f_c are easily implemented.

6.5.2 Envelope Control

Envelope control signals for each voice unit are generated by the play control module, concurrent with frequency control. On receipt of a "note turn on" command from the computer, the envelope attack control signal NTSTRT is generated and routed to the specified voice (see Figure 6.2). This applies to both percussive and non-percussive

envelope types. In addition, a voice status register is set. This register permits software interrogation of the instantaneous voice usage status (i.e. busy or free) and is required because of the time delay between the promulgation of envelope control signals and the completion of the envelope decay. On receipt of the NTSTRT control signal, the voice controller oversees the envelope clocking until the envelope steady-state is reached (non-percussive option), or until the envelope decay is complete (percussive option). In the latter case the voice controller sends a note "turn off" signal NTEND to the play controller, which inhibits the frequency clock (e.g. by resetting the pitch/voice multiplexer). The voice status register is also reset. With the non-percussive envelope option, a note "steady state ends, note decay starts" signal NTNDST is issued by the play software to initiate the envelope decay (see Figure 6.2). The NTEND signal is generated in the same manner and with the same effect as in the percussive envelope mode.

6.6 COMPUTER INTERFACE AND SOFTWARE CONTROL

The high-level interpretive software, with which the user interacts, is linked to the synthesiser hardware through the computer interface hardware and associated low-level control software modules which perform specific tasks such as voice load, note play control, and waveform specifier input.

Initially, the author's RTL digital interface hardware (see Section 3.3) was used, with the addition of

TTL line drive buffers and a data output synchronisation clock. In early 1976 a more comprehensive general-purpose interface (the BDI) was added to our EAI 640 digital computer as a peripheral device for general use. This was developed within the Electrical Engineering Department by W.K. Kennedy and F.M. Cady, and permits serial or parallel data transfers of up to eight 16 bit words, together with control words, data transfer pulses, and external hardware interrupts. Full details are given in the BDI User Manual which is lodged in the Department's Computer Laboratory. Once this BDI became fully operational the digital synthesis system together with the organ keyboard and conventional organ was transferred to it from the RTL interface, in accordance with departmental policy.

The current interface requirements for the digital synthesis system are as follows: one word (16 bits) output control, a half word (8 bits) output data, 24 bits input control, and 8 bits input data. In addition, a data output clock pulse is required for memory loading, and a general-purpose interrupt is used to provide clock timing and synchronisation between the waveform specifier hardware and its software. The organ keyboard and its associated play-back latches, used in the conventional organ of the music input/output system discussed in Chapter 3, require four 16 bit words of input and output respectively. While the output latches are not required by the digital synthesis system, the keyboard input is. To avoid unnecessary duplication of cabling, the conventional organ interface has been incorporated also. A further 20 bits input control

have been allocated for timbre control switches, and the future implementation of keyboard touch dynamics requires an additional 8 to 12 bits. Future output control requirements are likely to include a word each for continuous frequency and individual voice gain control. These I/O interface requirements are summarised in Table 6.2. The mode and status control bit allocations are listed with comments in Table 6.3.

A comprehensive suite of software modules is provided to control basic tasks such as loading a voice memory, reading the waveform specifier, checking the mode status, and starting and stopping notes of the equally-tempered scale. A preliminary version of a continuous frequency playback facility is also available, using a manually- rather than a computer-controlled oscillator for frequency generation. These modules are FORTRAN compatible, but are written in ASSEMBLY language for speed and because of the easy bit manipulation facilities provided in the latter. In addition to these hardware control routines, a number of companion utility modules are provided to generate analytical functions (see Section 6.7), display stored waveshapes and envelopes on the graphical display unit, smooth time functions using low-pass digital filtering, and to store and retrieve disc data files.

Software conventions used throughout are as follows:

(i) Waveshape and envelope data is stored in an integer array, 64 words long, containing 128 eight bit samples packed 2 samples per word.

(ii) A logical variable ENWAVE is .TRUE. if the data

is to be interpreted as a waveshape, .FALSE. if envelope.

(iii) A logical variable PERCUS is .TRUE. if percussive envelope option is required, .FALSE. if non-percussive.

(iv) A 3 word NAME array (packed, 2 ASCII characters per word) is used for file and plot identification.

(v) An integer VOICEN ranging from 1 to 16 is the current voice specified (for load or play). Hardware will ignore commands to non-existent voices (cf. hardware voice address convention - Table 6.3).

(vi) An integer PITCH is used to specify pitches of the equally-tempered scale. The software convention follows that of Section 3.4 - thus 0 is silence, and 16 to 76 correspond to notes C2 to C7 respectively (cf. hardware pitch address convention - Table 6.3).

Full documentation of the software is held in the Electrical Engineering Department.

6.7 WAVEFORM SPECIFICATION

It is envisaged that during normal use in either organ or synthesiser mode, the composer or performer will retrieve computer-stored timbres filed in a "sound catalogue", using a concept similar to that pioneered by Mathews and others with MUSIC V and GROOVE (Mathews, 1969; Mathews, Moore and Risset, 1974). To aid experimentation and permit extensions to the sound catalogue, facilities have been provided to expedite the specification of signals in either the time or frequency domains. Synthesis software permits the rapid generation of signals based on square or

triangle waveforms, or by Fourier synthesis from a discrete spectral representation using amplitude, frequency and phase parameters. A hardware "specifier" unit permits envelopes and waveshapes to be drawn on a perspex tablet, and subsequently sampled and digitally coded for computer storage or direct loading into the required voice memory.

Synthesis software currently available generates three types of functions which may be readily defined using few parameters. The first two types use rectangular and triangular base functions respectively, with duty factor from 0 to 100% as the controllable parameter. Thus pulses ranging from impulses to continuous dc levels, and triangular, saw-tooth and ramp functions may be rapidly generated. An additional feature which could be easily incorporated is the inclusion of zero-crossing parameters. Thus, Walsh-like functions could be generated, with the existing periodicity parameter corresponding to time base, and sequency as the additional parameter (Harmuth, 1972). The third function generation technique which has been implemented is Fourier synthesis, in which the signal $s(t)$ is calculated from

$$s(t) = \sum_{i=1}^{N} A_i \cos 2\pi(f_i t + \phi_i) \qquad (6.15)$$

over the interval

$$0 \leq t < \frac{1}{f_1} \qquad (6.16)$$

where the amplitude, frequency and phase parameters A_i, f_i and ϕ_i are specified. The use of the interval defined by

equation (6.16) as the normalised period permits the generation of functions with both harmonic and sub-harmonic components, as well as functions whose derivatives are continuous or otherwise. The Fourier synthesis facility has proved particularly useful, since it permits ready transformation between the frequency and time domains, and provides an efficient method of generating complicated timbres.

The function generation software normalises the output signal to the magnitude range 0 to $(2^8 - 1)$, required for an 8 bit representation with minimum quantisation error. Automatic display is incorporated, and a data filing facility permits the generated signal to be stored on disc if desired.

The waveform specifier hardware was developed by Vaughan (1975) as part of the initial undergraduate project. It consists of a sliding perspex tablet on which the desired (single valued) function is drawn. The tablet is "read" by sliding it between a light source and a bank of 64 photo-transistors. A separate clock or sample control photo-transistor is used to ensure that function sampling occurs at uniform spatial intervals, by synchronising the sampling to a timing track incorporated on the tablet. Thus, at each sampling position the clock phototransistor is activated, initiating a rapid scan of the sampling phototransistor array. The 6 bit code corresponding to the address of the first blanked phototransistor is the current sample magnitude. In this manner 128 six bit samples are generated, with a linear coding characteristic and spatially

uniform samples independent of the tablet sliding velocity. Specifier mode control switches are used to define whether the function drawn is a waveshape or an envelope, and whether it is to be read into the computer or directly into a voice memory (see Table 6.3). To facilitate the latter, a pair of thumbwheel switches is provided to generate the destination voice address.

Software smoothing facilities are provided to interpolate the 6 bit magnitude to the normalised 8 bit form. This is currently achieved by digital low-pass filtering.

6.8 CONCLUSION

The digital synthesis system described in this chapter permits a wide range of sound timbres to be generated in real-time under manual or pre-programmed computer control. The sound generation model is formulated in a manner which permits the individual contributions of periodic waveshape, envelope, tremolo and vibrato to be assessed. Since the actual sound generation is performed by digital hardware rather than software (cf. MUSIC V), real time operation is possible and the computer is free to perform supervisory and interpretive tasks. Another significant factor in favour of using dedicated hardware rather than software for sound production is that high speed mass data storage is not required. Thus the system does not require high speed digital magnetic tape (10 to 20 kHz word access rate) or disc storage with DMAC facilities - neither of which is available on our EAI 640

computer.

The advantages of digital synthesis over analogue synthesis techniques are well known, and include more precise control, greater accuracy, stability and repeatability, as well as the facility for generating arbitrary time functions. Moreover, digital hardware is amenable to time sharing so that hardware parallelism may be avoided.

The system described above was designed and implemented on a modest budget. While it is a continuing project and still not complete, the facilities available at present permit an impressive range of tones and sound colourations to be generated. The system has already proved to be a useful extension of the music output system discussed in Chapter 3, both for performance and as a composition aid.

Vaughan (1977) provides detailed documentation of the existing system hardware.

TABLE 6.1

DIGITAL GENERATION OF THE EQUALLY TEMPERED SCALE

Pitch	Division Factor	Frequency (Hz)	Error (%)
C7	239	2 092.1	0.04
B6	253	1 976.3	0.04
A#6	268	1 865.7	0.06
A6	284	1 760.6	0.03
G#6	301	1 661.1	0.01
G6	319	1 567.4	0.04
F#6	338	1 479.3	0.05
F6	358	1 396.6	0.02
E6	379	1 319.3	0.06
D#6	402	1 243.8	0.06
D6	426	1 173.7	0.09
C#6	451	1 108.6	0.01

TABLE 6.2

CONTROL AND DATA INPUT/OUTPUT INTERFACE REQUIREMENTS

INPUT (To CPU)

24 bits	Control. (8 bits mode status, 16 bits voice status.)
8 bits	Signal Data. (From waveform specifier.)
60 bits	Keyboard Data. (1 bit per key, 5 octave range.)
20 bits	Timbre Control. (From manually operated switches.)
8-12 bits	Keyboard Touch Dynamics. (Proposed facility.)

OUTPUT (From CPU)

16 bits	Control. (Controller mode select.)
8 bits	Signal Data. (To voice waveshape or envelope memory.)
60 bits	Keyboard Latch Data. (Used in conventional organ playback.)
12-16 bits	Proposed Portamento Frequency Control.
12-16 bits	Proposed Voice Gain Control.

TABLE 6.3

MODE AND STATUS CONTROL WORD BIT ALLOCATIONS

MODE CONTROL (OUTPUT)

Bit Number	Function	Values
0	Load/Store. (Mode of memory specified from bits 1 and 4-7)	0 = Load 1 = Store
1	Envelope/Waveshape. (Multiplexer control for voice memory addressing, with bits 4-7)	0 = Envelope 1 = Waveshape
2	Playback Addressing Mode. (Keyboard Playback/Random Voice Addressing Playback)	0 = Keyboard 1 = Random
3	Equally Tempered Scale/Portamento (Applies only if bit 2 = 1). (Bits 2 and 3 are used to switch the waveshape clock input to the E.T.S. MUX output, or to the local frequency generator)	0 = E.T.S. 1 = Portamento
4-7	Voice Address (4 bits). Used in memory load, or random voice access playback.	0 - 15 = Voice address
8-13	Pitch Address (6 bits). Used for playback, random voice access mode, equally-tempered scale option. (Pitch-Voice MUX Control)	0 - 60 correspond to notes C2-C7. 61-62 illegal. 63 silence.
14	Pitch-Voice MUX enable, and strobe control for envelope timing. (NTSTRT = bit 14, if Pitch Address $\neq$ 63, NTNDST = bit 14, if Pitch Address = 63)	1 if note starting or ending, 0 otherwise.
15	Envelope Mode. Percussive/Non Percussive	0 = Percussive 1 = Non-percussive.

TABLE 6.3 (Continued 2)

MODE STATUS (INPUT)

Bit Number	Function	Values
0	Load/Store (Waveform specifier status, is "load" when specifier data ready, otherwise "store")	0 = Load 1 = Store
1	Envelope/Waveshape (Control console switch, used with waveform specifier)	0 = Envelope 1 = Waveshape
2	Computer/Manual (Control console switch, indicating waveform specifier load into computer or voice memory)	0 = Computer 1 = Manual
3	Envelope Mode. Percussive/Non Percussive (Control console switch, cf. bit 1)	0 = Percussive 1 = Non-percussive

TABLE 6.3 (Continued 3)

VOICE STATUS (INPUT)

Bit Number	Function	Values
0-15	Current Status of all voice controllers during playback. Bit 0 corresponds to voice 1, bit 15 to voice 16. Voices not implemented are held busy. This facility is required because of time delay between software command "turn note off" (non percussive) or "turn note on" (percussive) and envelope decay completion.	0 = busy 1 = not busy.

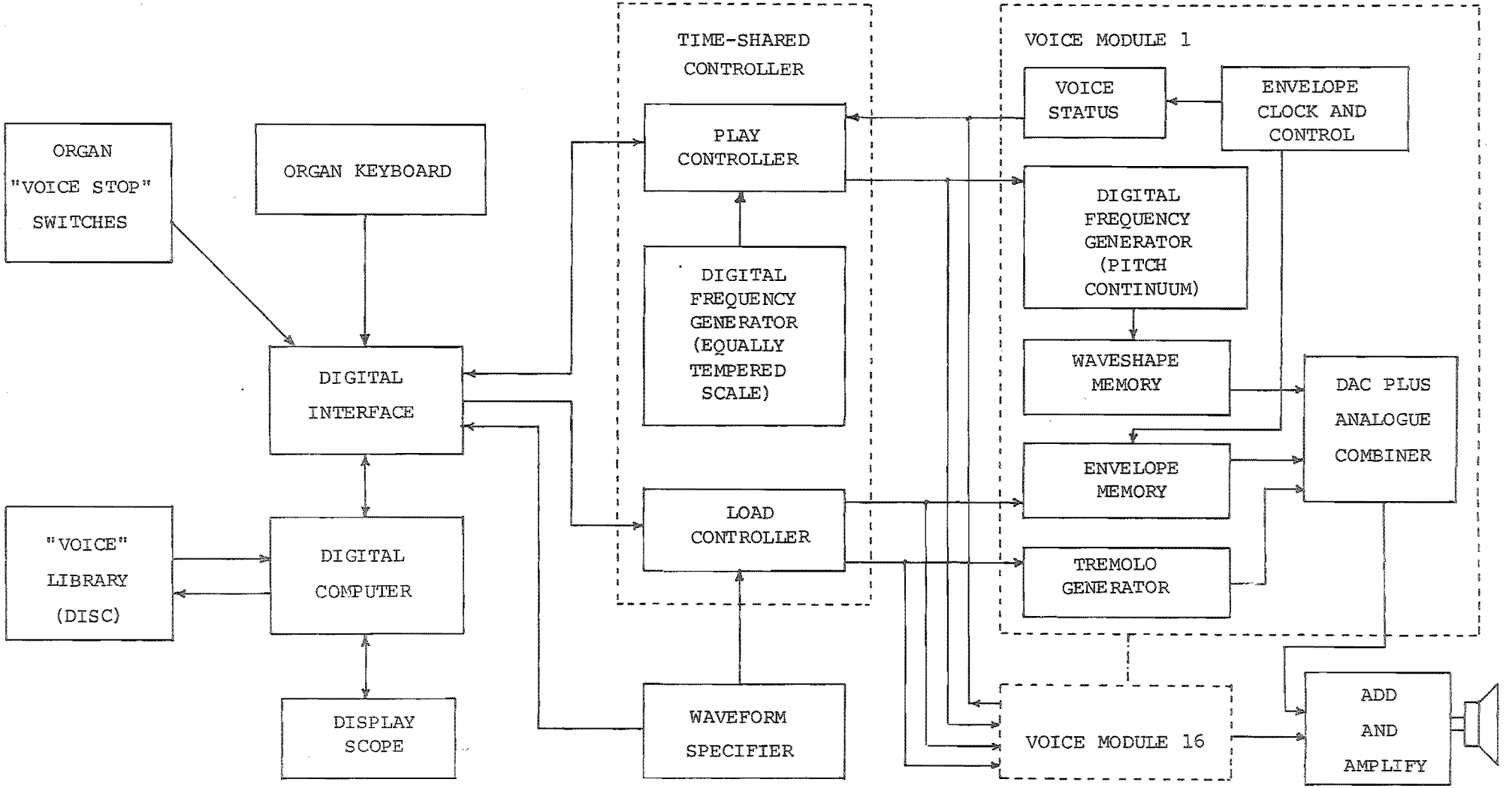
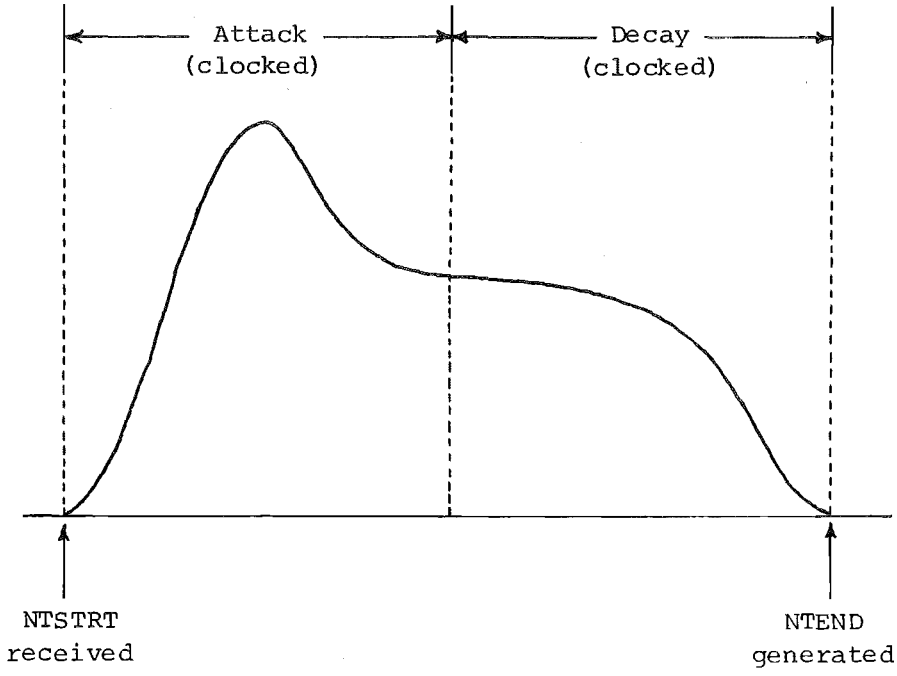


Figure 6.1 Block diagram of the Digital Synthesiser



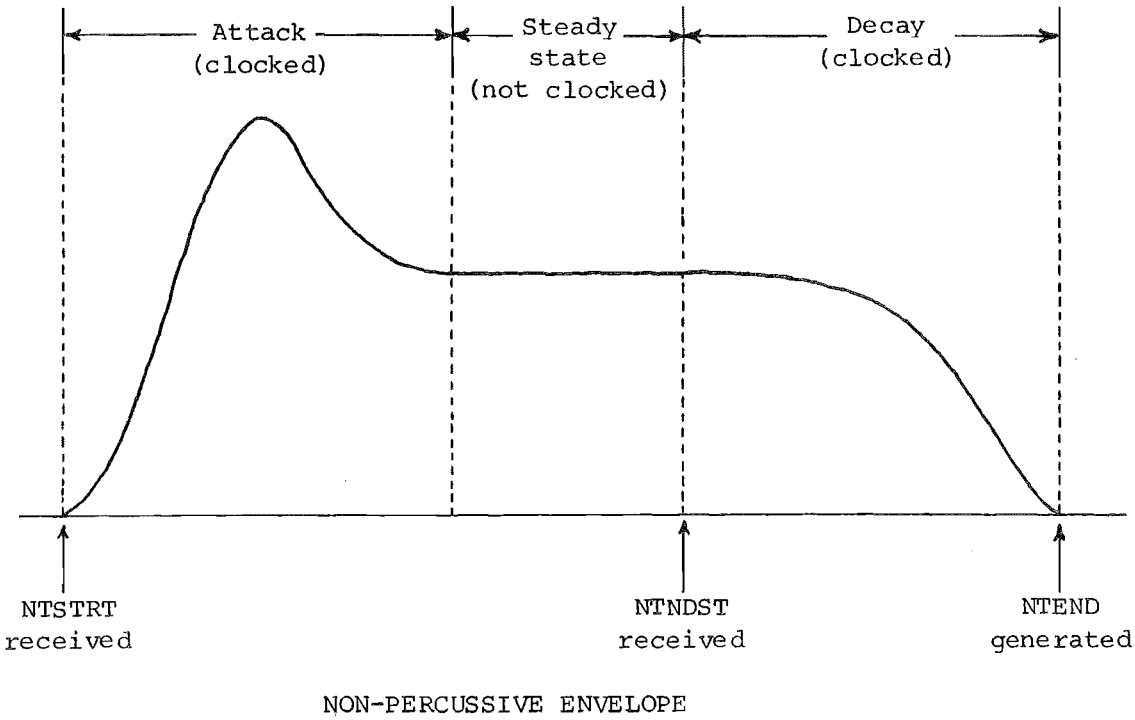


Figure 6.2 Percussive and non-percussive envelope generation.

PART 3

PITCH ESTIMATION IN SPEECH AND MUSIC

CHAPTER 7

PITCH ESTIMATION - A REVIEW

7.1 INTRODUCTION

The problem of pitch estimation in speech and music is not new. Pitch is a parameter fundamental to both, and theoreticians considering the analysis of speech or music sounds have long desired an objective technique for measuring pitch accurately. As with many instrumentation problems, the development of electronic technology offered the first tools for a practical solution. Pioneering work in speech pitch analysis was done by Scripture (1902, 1903, 1923), Meyer and Schneider (1913), Bien-Ming (1931), Hunt (1935) and Cowan 1936). In music, early workers include Metfessel (1926), Seashore (1932), Tiffin (1932), Grützmacher and Lottermoser (1937), Railsback (1937) and Obata and Kobayashi (1937, 1938). The approach generally used was to measure the fundamental frequency of the signal being analysed and to assume an isomorphic relationship between this measured frequency and the perceived pitch. Typical of the techniques used were those based on stroboscopic principles (cf. Railsback, 1937) and oscillator synchronisation (cf. Grützmacher and Lottermoser, 1937; Obata and Kobayashi, 1937, 1938). In the stroboscope method, a neon lamp driven by the amplified signal is used to illuminate a stroboscopic screen on a drum moving at

constant speed. This approach is still widely used in orchestral frequency meters for measuring intonation, and Backus (1970) states that accuracy to about one cent (0.01 semitone) is achievable. Unfortunately it is not suitable for automation. With oscillator synchronisation, the signal under analysis synchronises a free-running oscillator whose pulses are used as input to a conventional frequency measuring device. Implicit in the latter and many subsequent techniques is the concept of preprocessing to enhance the energy content of the signal fundamental relative to the total signal energy, prior to fundamental frequency measurement.

Other preprocessing techniques which have been used include filtering (either with fixed, manually tuned or automatic tracking filters), and non-linear time-domain transformations followed by filtering. For the many frequency meters which operate in the time domain by measuring the rate of zero crossings (McKinney, 1965; Tove, Norman, Isaksson and Czekajewski, 1966), the requirement on the preprocessor is that its output possess only two zero-crossings per signal period. McKinney (1965) presents a relationship between the fundamental and harmonics of a signal which he hypothesises is a necessary and sufficient condition for this zero-crossing requirement. He also investigates the effect of squaring, cubing, full and half-wave rectification, and Gaussian and logarithmic transformations to achieve this condition. Since these techniques are well documented by McKinney, and since none is universally acceptable, these "classical analogue"

methods are not described herein.

The primary motivation of early researchers appears to be instrumentation for analytical studies of linguistics or music. Following Dudley's (1939) invention of the Vocoder, a significant change of motivation occurred, and research effort since the early 1950's has been strongly polarised towards the search for an automatic, reliable and accurate speech pitch estimator for use in the analysis portion of vocoder systems. This motivation is prompted by the (as yet largely unrealised) commercial possibilities of analysis-synthesis telephony systems for speech bandwidth reduction, and is reflected directly or implicitly in the pitch estimation literature (McKinney, 1965).

A consequence of this research emphasis is that pitch estimation techniques have become very specialised and are directed almost exclusively towards speech signal analysis. Thus, *a priori* knowledge of the speech signal characteristics is implicitly assumed in many recent techniques - for example cepstral analysis (Noll, 1964) and inverse filtering (Markel, 1972b) assume the frequency separability of the source excitation and vocal tract response functions of speech. As shown in Section 7.3, these assumptions are not in general valid for many music signals. Also, while the desired operational frequency range of most speech pitch estimators is about a decade (from 50 Hz to 500 Hz if both male and female speakers are permitted), in music a frequency range of 75 (about 40 Hz to 3000 Hz is desired. (Backus, 1970), specifies a 7 octave range from 27 Hz to 4200 Hz.) Thus, techniques such as optimum comb filtering

or autocorrelation analysis, whose computational complexity depends on the square of the signal sampling frequency (Rabiner, Cheng, Rosenberg and McGonegal, 1976) become very unwieldly if the frequency constraints are extended to include the desired pitch range for music.

The motivation for the work in this part of the thesis was the desire to implement economically a pitch estimator suitable for as wide a class as possible of single-voiced (unisonous) instruments, including the voice singing or humming. This pitch estimator then forms an alternative to the keyboard music input system described in Part 1, and significantly enhances the flexibility and versatility of the overall system. While ultimately desirable, the extension from unisonous to multisonous instruments (or to the composite signal of a group of unisonous instruments) has not been considered. As shown in Chapter 9, the restriction to unisonous instruments is not as severe as it initially appears. Most non-keyboard instruments are unisonous, and the use of compact, light-weight motion-sensitive transducers connected to individual instruments permits the signal from each instrument in an ensemble to be available with high signal-to-noise-ratio.

This chapter presents a detailed review of many recent techniques which have been applied to speech pitch analysis, and discusses their suitability in a music-oriented application in the light of the preceding comments. Section 7.2 discusses briefly pitch perception and the relationship between the physical parameters of a signal and the perceived pitch. The pitch estimation problem is

defined, and a general specification of the "ideal" pitch estimator for both speech and music is given. This is followed in Section 7.3 by a comparison of objective features of speech and music signals. The techniques reviewed can be categorised as follows: methods based on auto-correlation, with or without signal preprocessing (Section 7.4), single and double spectrum analysis methods (Section 7.5), linear prediction and inverse filter techniques (Section 7.6), pitch estimation from measurements of glottal movement (Section 7.7) and heuristic methods (Section 7.8). Most of these techniques have been developed or expanded since McKinneys' (1965) detailed review of speech pitch estimation methods. The widespread use of general-purpose digital computers, and more recently the emergence of high-speed special-purpose digital signal processors has been largely responsible for the change in approach prevalent in McKinneys' review to that presented here. Also the formal introduction and subsequent widespread use of the fast Fourier transform in 1965 has played a significant role in changing attitudes away from the "classical analogue" approach to the current digitally-oriented approach.

7.2 PITCH AND FREQUENCY

7.2.1 Pitch Perception

The human pitch perception mechanism is not yet well understood, either from a perceptual or "black box" viewpoint (behavioural psychology) or in terms of the physiology of the auditory pathway (Flanagan, 1972).

It is outside the scope of this thesis to review the extensive literature on pitch perception - the intention is merely to outline the main factors involved so that the pitch estimation task may be better appreciated. For a comprehensive review, the reader is referred to Plomp and Smoorenburg (1970).

Pitch is difficult to define. Flanagan (1972) has qualitatively described pitch as "that subjective attribute of auditory sensation that admits of rank ordering on a scale ranging from low to high." In general, for periodic waveshapes, a strong correlation exists between the perceived pitch and the signal periodicity.

Various factors influence the perception of the pitch of a complex waveform - masking of certain spectral frequencies by other nearby components, beating between spectral frequencies, the generation of combination tones by non-linearities in the auditory mechanism, and the non-linear relationship between perceived loudness and frequency. Because of such factors, it is possible to devise periodic sounds for which the pitch is uncertain by one or more octaves (Shepard, 1964) or for which several distinct pitches may be clearly heard (Schouten, Ritsma and Cardozo, 1962). In addition, non-stationary waveforms (whose spectral components are inharmonic) - for example bells, gongs and timpani - may produce a definite pitch sensation, although the pitch is not the same as that of a pure tone of the same fundamental frequency (Mathews, 1969).

While these facets of pitch perception are important for an understanding of the auditory system, they are of

less significance in normal musical listening behaviour. As Bilsen (1973) has pointed out, in normal musical listening one does not listen analytically - i.e. one does not attempt to distinguish the component partial tones, even though this is possible provided the components are spaced "sufficiently widely" (Schouten *et al.*, 1962). The overall non-analytical pitch sensation which is important in normal listening has been shown to be effected by the periodicity of the sound waveform, rather than by a spectral (Fourier) transformation of the waveform to extract the fundamental frequency. This is easily demonstrated by considering a periodic waveform for which no energy exists at the fundamental frequency - for example telephone speech, music reproduced by small transistor radios, or notes played by certain instruments such as the bassoon. This "periodicity" pitch is called residue pitch (Schouten, 1940), periodicity pitch (Licklider, 1954), repetition pitch (Bilsen and Ritsma, 1970), musical pitch (Houtsma and Goldstein, 1972) or virtual pitch (Terhardt, 1972) to distinguish it from "place" pitch. The latter is evoked by the fundamental (Fourier) component of the waveform and is physically present as a resolved entity in the vibration pattern of the basilar membrane (Flanagan, 1972).

Periodicity pitch suggests that the auditory pitch extractor operates in the time domain and uses a process similar to autocorrelation, rather than in the frequency domain (for example measuring the spacing of spectral components) (Bilsen and Ritsma, 1970). Schouten *et al.* (1962) comment that the time-domain hypothesis "gathered

momentum during the last twenty or so years on account of both perceptual and anatomical evidence", and cite Licklider (1951, 1956), Whitfield (1957) and Goldstein (1957). Recent experiments by Houtsma and Goldstein (1972) show that pitch cannot be a result of autocorrelation at one single place in the peripheral auditory pathway, since dichotic presentation of a two-component stimulus (i.e. one frequency component is presented to one ear and the adjacent harmonic to the other) evokes essentially the same pitch sensation as a monotic presentation. In consequence it is concluded that pitch is extracted at higher centres in the auditory pathway, and that significant neural processing is involved.

7.2.2 Pitch Discrimination

Man is highly sensitive to differences in the frequency of tones presented for comparison. Rosenblith and Stevens (1953) report that under certain conditions the threshold for detecting a difference in the frequencies of two pure tones presented successively is as small as one part in 1000. Backus (1970), citing Shower and Biddulp (1932) observes that for frequencies up to about 1000 Hz the ear can detect changes of about 3 Hz, and comments that this explains the poor pitch discrimination at the low frequency end of the hearing range. At frequencies above 1000 Hz the pitch discrimination remains approximately constant at about 0.25% or 0.04 semitones. These figures are confirmed by Flanagan (1972) who states that for synthetic vowel sounds the fundamental-frequency limen is about 0.3% to 0.5% of the fundamental frequency.

While difference limens - or just noticeable differences - are of interest for psycho-acoustic research because they set an upper bound on the resolving ability of the perceptual mechanism, their use as a fidelity or performance criterion is misleading, as Flanagan (1972) points out. Real-world listening tasks, such as those encountered in normal speech or music perception, involve absolute as well as differential judgements of multi-dimensional sounds for which loudness and duration perception is also important. Under these conditions, pitch discrimination is reduced (Pollack, 1952; Pollack and Ficks, 1954; Flanagan, 1972).

The subjective pitch scale, whose unit is the mel which is defined by a tone at a frequency of 1000 Hz, is widely used in psycho-acoustic research (Backus, 1970). Since this is of little significance to speech or music it is not used in this thesis. The natural unit for a musical scale is the octave and unless stated otherwise, the equally-tempered scale (cf. Section 9.7) is used herein.

7.2.3 "Pitch" Measurement

As mentioned in Section 7.2.1, a strong correlation generally exists between the perceived pitch of a periodic waveform and the signal periodicity. The basic objective of pitch estimation is to measure this periodicity, either by direct time domain techniques such as feature recognition, by frequency domain methods such as cepstral analysis, or by extracting the excitation signal (in speech, separating the glottal pulses from the vocal tract impulse response, for

example using inverse filtering techniques). The widely accepted "ideal" standard which is used for comparison of different schemes or where extreme accuracy and reliability is required, is "eye" detected pitch (McKinney, 1965; McGonegal, Rabiner and Rosenberg, 1975). "Eye" detected pitch is obtained by application of human pattern recognition faculties to a visual display of the signal. While McKinney has excluded the use of human pattern recognition on functions derived from extensive modification of the acoustic signal (e.g. narrow-band spectrograms or other spectral representations), McGonegal *et al.* use a composite display in which the low-pass filtered signal display is complemented by the short-time autocorrelation and cepstrum functions.

While for pitch analysis it does not in principle matter where the "beginning" of each "pitch period" is defined to occur, provided it is consistent, a widely used definition is "coincident with the zero crossing before the principal peak in the period" (Mathews, Miller and David, 1961; Atal, 1968; Miller, 1975). McGonegal *et al.* (1975) do not restrict their definition to the principal peak, presumably because of a problem in speech signals similar to the "hop" mentioned by Gold (1962a), and pointed out by McKinney. This problem arises when gross changes in the speech waveshape occur over several pitch periods (for example during fast vocal tract transitions). However, McGonegal *et al.* still use pitch period markers positioned at (positive going) zero crossings where the signal slope is high and where positional uncertainty due to high

frequency noise is low.

Another suitable pitch period marker is the principal peak (Gold and Rabiner, 1969). Rabiner, Cheng, Rosenberg and McGonegal (1976) observe that both zero-crossings and peaks are suitable for pitch period markers, but that speech period measurements based on zero crossings will in general differ slightly from those based on peaks. These pitch period discrepancies are due not only to the quasi-periodicity of the speech waveform, but also to the fact that peak measurements are sensitive to the formant structure during the pitch period, whereas zero crossings are sensitive to the formants, noise and any dc or very low frequency component. The author's experience suggests that period measurements based on peaks are more consistent than those based on zero crossings (cf. Section 9.2). Figure 7.1 illustrates these period marker definitions.

Another approach to the extraction of pitch period markers in speech is "epoch extraction" (Dolanský, 1955; Lerner, 1959; Anderson, 1960; Ananthapadmanabha and Yegnanarayana, 1975). The underlying assumption is that a discontinuity exists in one of the speech signal derivatives. The point of discontinuity of the lowest-ordered derivative is an epoch point, and is used as a pitch period marker. Physically, the epoch can be considered as a point of glottal pulse excitation of the (resonant) vocal tract (cf. Section 7.3). However, since the discontinuity sometimes occurs only in the high-order derivatives, this approach is susceptible to noise. Recent work (Ananthapadmanabha *et al.*, 1975) also indicates that several

markers can be derived in one pitch period.

Real speech and music signals are dynamic and exhibit temporal variations which must be reflected accurately in the pitch estimator output. Note start and end times (in music), and the onset and end of voicing (in speech) should be resolved within, typically, several pitch periods or at least 10 to 20 ms. This "time event" resolution requirement is additional to the necessity for accurate tracking of temporal variations of pitch period - such as vibrato and glissandi in music, or stress and intonation in speech. This leads to the concepts of "instantaneous pitch" and "pitch trajectory". "Instantaneous pitch period" is here defined as the time interval between the most recent two pitch period markers. "Pitch period trajectory" is the instantaneous pitch period as a function of time. Using these definitions, the pitch period trajectory is (in general) piecewise constant, with jump discontinuities at each point in time when pitch period markers occur.

It is worth pointing out that the pitch period trajectory may be sampled (or estimated) once every pitch period (i.e. pitch synchronously) or at uniformly spaced time intervals (pitch asynchronously).

7.2.4 A note on Terminology - "Pitch" Clarified

Terminology in the pitch estimation literature can be misleading, since the terms "pitch" (a subjective phenomenon) and "fundamental frequency" or "F_o" (an objective property) are often used interchangeably. To compound confusion, periodic signals may have little or no energy at a spectral frequency corresponding to the "fundamental" (or

repetition) frequency. In speech analysis, this confusion can be eliminated by specifying the "laryngeal frequency" (cf. McKinney, 1965), although this is not applicable to other sounds. Ward (1954) suggests that the terminology "pitch level" or "pitch equivalent" with units of Hz be used to denote the repetition frequency. While this suggestion has not gained acceptance there is considerable merit in the use of terminology which explicitly distinguishes subjective pitch and its physical correlates. In this thesis the term "pitch frequency" is used to denote repetition frequency if there is any doubt whether this or the subjective pitch is intended. However, terms such as "pitch estimator" and "pitch trajectory" are firmly entrenched in the literature, and there is little point in abandoning accepted usage.

Some writers prefer to deal with the "pitch period" of sounds, since no ambiguity exists and since the notion of repetition is implied in "period". Because pitch period and pitch frequency are reciprocals, both descriptions are equivalent.

7.3 SPEECH AND MUSIC COMPARED

Speech and music signals are only approximately periodic. Departures from periodicity result from noise, and from perturbations of the excitation or response of the sound-producing mechanism. These latter cause the signal to vary both in period and in the detailed structure of the waveform within a period.

During voiced speech, significant changes in the waveform frequently occur within several pitch periods. This applies especially during fast vocal tract transitions, when the formants are changing rapidly. This non-stationarity is compounded by interaction between the vocal tract and the glottal excitation. Under some conditions the formants of the vocal tract can alter significantly the structure of the glottal waveform (cf. Flanagan, 1972). Such interactions occurring during rapid formant changes are particularly deleterious to pitch estimation (Rabiner, Cheng, Rosenberg and McGonegal, 1976). Figure 7.2(a) illustrates a typical rapid transition, in this case the voiced plosive /d/ .

While speech can generally be regarded as stationary over only a 10 to 20 ms interval, some vowels may be essentially stationary for up to 100 ms. Figure 7.2(b) shows the vowel /u/ , and illustrates the highly resonant nature of voiced speech. Four formants (resonances) are typical in the band 0 to 5 kHz.

The voiced/unvoiced transition which is peculiar to speech provides additional complications for pitch estimation. Transitions between unvoiced speech and low level voiced speech are often subtle, and consequently difficult to pinpoint. Such a transition is illustrated in Figure 7.2(c). Another complication which is peculiar to speech is a condition called "diplophonia" (Flanagan, 1972), in which alternate glottal pulses are of different amplitude. This produces a waveform for which alternate periods are highly correlated, but adjacent periods are poorly

correlated. An example of diplophonic speech is given in Figure 7.2(d), in this case the vowel /a/ .

In contrast to the dynamic nature of speech, the signals produced by most non-percussive instruments are characterised by an attack transient, a relatively long steady-state portion during which the signal is essentially stationary, and a decay segment. This is true also for humming. The attack transient incorporates both a temporal envelope variation and inter-period changes in the waveform - these latter are due to energy transfers between different modes of excitation. Such modal energy transfers are illustrated in Figure 7.2(e), which is a segment of the attack transient of a bassoon note. In this case the pitch doubles as a result of the energy transfer. The attack transient duration depends on the fundamental frequency, and is longer for low-pitched notes than for high (Strong and Clark, 1967a). Nevertheless it is usually in the range 10 to 100 ms, or about 10 to 20 pitch periods (Freedman, 1967; Keeler, 1972).

The decay transient is usually of similar duration to the attack transient, but is characterised more by a decreasing temporal envelope than by significant changes in the waveshape. The steady-state signal may be perturbed by frequency and amplitude modulations caused by vibrato and tremolo, respectively. These modulations are usually small, and seldom exceed 5% in frequency variation, or 20% in amplitude. Consequently the musical signals of interest are of similar shape between adjacent periods, over most of the signal duration.

A complication encountered with speech signals in some applications should be mentioned here. When speech is transmitted through the telephone system, the signal suffers linear filtering, non-linear processing, and the addition of noise (Rabiner, Cheng, Rosenberg and McGonegal, 1976). The telephone system acts as a band-pass filter, retaining the frequency band from about 200 Hz to about 3200 Hz. This can significantly attenuate the fundamental pitch frequency and many of the higher pitch harmonics, making the periodicity more difficult to measure. Non-linear effects include phase distortion, fading or amplitude modulation of the signal, crosstalk between several messages, and clipping of very high-level sounds. While these effects are not considered in this thesis, they should be borne in mind for those applications which must process telephone-quality speech.

7.3.1 Speech Generation

The human speech production mechanism can be modelled as a source and filter, where the output speech signal is the convolution of the vocal tract impulse response with the source waveform. The vocal tract is a system of resonant cavities (the mouth and nasal passages) which can be excited in several ways. For voiced sounds the source is a pulse train produced by the vocal cord oscillator. Unvoiced sounds are generated either by passing an air stream through a constriction in the vocal tract (e.g. the fricative consonants /f/,/θ/,/s/,/ʃ/), or by making a complete closure, building up pressure behind the closure and abruptly releasing it (e.g. the plosive or stop

consonants /p/,/t/,/k/). The former results in turbulent flow and incoherent sound, while in the latter a brief transient excitation of the vocal tract occurs. The fricative and stop consonants can also be produced in conjunction with voicing. Examples of the voiced fricative consonants are /v/,/ð/,/z/,/ʒ/ while the voiced stop consonants include /b/,/d/ and /g/.

The non-nasalised vocal tract may be represented by an all pole model. Nasalisation (which is controlled by the velum) introduces zeros into the filter transfer function (Flanagan, 1972). However, as Atal and Hanauer (1971) point out, these zeros can usually be modelled by multiple poles. Consequently an all-pole filter model is frequently used for both speech analysis and synthesis.

The source-filter model for speech assumes that the excitation and vocal-tract response are essentially independent - i.e. that any coupling between them is small. Since the acoustic impedance of the glottal source is usually large compared to the acoustic impedance looking into the vocal tract, this assumption is often correct. Nevertheless significant interaction can occur when the vocal tract is tightly constricted (Flanagan, 1972), as already noted.

Sundberg (1977) discusses the singing voice, and describes the effect of the lowering of the larynx and the expansion of the pharynx and laryngeal ventrical which is peculiar to the articulation of singers. This has the effect of creating an additional formant - designated the "singing formant" - which lies between the third and fourth

formants of normal speech.

For a comprehensive review of the speech production mechanism and the many models which have been formulated, the reader is referred to Flanagan (1972).

7.3.2 Music Generation

Musical instruments are conveniently categorised as percussive or non-percussive, depending on whether the excitation is impulsive or periodic. Because of the transient waveform which results from most percussive instruments (Backus, 1970) only non-percussive instruments are considered in this thesis. Thus the percussion instruments (such as timpani), piano, harp and other plucked string instruments are excluded, but the brass, woodwind and bowed string instruments are considered herein.

The woodwind instruments consist of a tube which is approximately cylindrical or conical. Openings in the tube wall permit the well-defined and harmonically-related resonance frequencies of the tube to be altered. These openings are also the points from which most of the sound energy is radiated. Excitation is achieved using edge tones (for the flute and recorder) or a vibrating reed (e.g. clarinet and oboe). In contrast with speech the excitation and response are tightly coupled, and significant interaction occurs. Thus, the vibration of the reed in a clarinet follows closely the internal pressure variations within the mouthpiece, but lags slightly because of the damping effect of the player's lip (Backus, 1970). This causes the actual playing frequency to be slightly below the resonance frequency of the clarinet bore. Since this frequency shift

depends on the average gap between the reed tip and the mouthpiece, the player can adjust the playing frequency by up to about ± 0.2 semitone (Backus, 1970). Witten (1977) has pointed out that for some notes, lip control can produce frequency variations which are much greater than those suggested by Backus.

The brass instruments are acoustically similar to the woodwind, although they differ in several important respects. The air column within the instrument is excited by the player's lips. Since these are comparatively massive (cf. the light woodwind reed) they can substantially influence which of the possible modes of vibration of the air column is excited. Since the lip motion is approximately sinusoidal, the excitation contains relatively few harmonics (Martin, 1942), unlike the reed excitation which can contain many harmonics (Backus, 1961).

Because of the ability of the lip vibration frequency to influence which modes are excited, numerous notes can be played by selecting the appropriate mode (e.g. bugle). In addition, the player can adjust the playing frequency of a particular mode by as much as 0.75 semitone (Backus, 1970). Since the largest interval between usable modes is the fifth, the production of a complete scale requires six additional semitones. These are generated by altering the tube resonance frequencies, which is achieved by lengthening the air column (cf. the woodwind, which use openings in the tube wall). Since there exist no wall openings in the brass instruments, all the sound is radiated from the bell. This permits the tonal quality to be significantly altered using

mutes (unlike the woodwind, for which mutes have little effect).

The string instruments consist of several (usually four) tensioned strings mechanically coupled to an acoustic amplifier. A string is periodically excited at its resonance frequencies by bowing, which results in an approximately sawtooth motion in the string at the point of bowing. In consequence, the string oscillation contains numerous harmonics, whose relative amplitudes depend on the position of bowing, the bow speed, and the bowing pressure. The resonance frequencies of the string are altered by adjusting the effective length and/or tension of the string (although in normal playing the latter is usually held constant).

The string vibrations are transmitted to the sound box amplifier through the bridge. The sound box has a complicated vibration spectrum, which exhibits numerous resonance frequencies in addition to the single resonance caused by the enclosed air.

A characteristic common to all the instruments discussed here is the comparatively tight acoustic coupling between excitation and response. This interaction makes untenable for musical instruments the source-filter model of sound production which is widely used for speech, and has widespread implications in the pitch estimation task.

Backus (1970) provides a detailed discussion of sound production by musical instruments.

7.4 METHODS BASED ON AUTOCORRELATION

7.4.1 Autocorrelation Analysis

Autocorrelation analysis is an old and well established technique for extracting the periodicity of quasi-periodic signals, or of periodic signals embedded in noise (cf. Papoulis, 1962; Lathi, 1965). Consider an infinite periodic signal $s(t)$ of periodicity T. Then the average autocorrelation function $\bar{\phi}(\tau)$ is defined by

$$\bar{\phi}(\tau) = \lim_{\xi \to 0} \frac{1}{\xi} \int_{-\frac{\xi}{2}}^{\frac{\xi}{2}} s(t)\, s(t+\tau)\, dt$$

$$= \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s(t)\, s(t+\tau)\, dt \,. \qquad (7.1)$$

Since $s(t)$ has the period T, it follows that

$$s(t) = s(t+T) \,. \qquad (7.2)$$

Hence

$$\bar{\phi}(0) = \bar{\phi}(T) = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} s^2(t)\, dt \qquad (7.3)$$

so that $\bar{\phi}(\tau)$ is itself periodic with period T. The term "average autocorrelation" is used here (cf. Lathi, 1965) because of the unbounded energy of the infinite periodic signal. For real-world signals which are neither infinite nor of unbounded energy, a more convenient definition of the autocorrelation function is (Lathi, 1965)

$$\phi(\tau) = \int_{-\infty}^{\infty} s(t)\, s(t+\tau)\, dt \,. \qquad (7.4)$$

Note that $\phi(\tau)$ is an even function, and that $\phi(0)$ corresponds to the total signal energy.

If $s(t)$ is zero outside the interval $t_1 < t < t_2$, and periodic, T, within this interval, then $\phi(\tau)$ is similarly periodic, T, but weighted by a linear taper which is maximum at $\tau = 0$ and zero at $\tau = \pm (t_2 - t_1)$. Hence $\phi(\tau) \leq \phi(0)$. This superimposed linear taper results from the convolution of the multiplicative rectangular window applied to $s(t)$, and simplifies the measurement of T. Thus, T is obtained by locating the largest peak of $\phi(\tau)$, other than the peak at $\tau = 0$. Then T equals the value of τ corresponding to this peak. For $s(t)$ to satisfy the periodicity requirement, the rectangular window $[t_1, t_2]$ must encompass at least two complete periods.

If $s(t)$ is only quasi-periodic, and if in addition the window $[t_1, t_2]$ spans many periods so that

$$T \ll t_2 - t_1 \tag{7.5}$$

and the effect of the linear taper is small, then the ratio $\phi(T) / \phi(0)$ provides a measure of the inter-period waveshape correlation within $s(t)$. This criterion is often used to make a voiced/unvoiced decision in speech analysis, even with short-time analysis when the condition (7.5) is not obeyed (cf. Dubnowski, Schafer and Rabiner, 1976).

The discussion has so far been concerned with real stationary signals. Most of the speech and music signals of interest are non-stationary. To observe the temporal variations in the signal periodicity, a running auto-correlation function is more appropriate. A variety of

definitions are given by Fano (1950) and Schroeder and Atal (1962), each oriented towards different methods of implementing the autocorrelator. Since the practical difficulties encountered with this early work are largely circumvented when digital rather than analogue techniques are used, these definitions are not discussed here.

The essential idea behind running (or short-time) autocorrelation analysis is that the signal $s(t)$ is multiplied by a suitable window. The autocorrelation function of the resulting windowed signal is evaluated. The window is then advanced in time, and the analysis proceeds. In this way real time t is introduced as a secondary variable in the autocorrelation function. Thus, for a rectangular window of length W, the short-time autocorrelation function is defined by

$$\phi(\tau,t) = \int_{t-W}^{t} s(x)\ s(x+\tau)\ dx \quad . \tag{7.6}$$

Short time autocorrelation analysis forms the basis of numerous pitch estimators discussed in this section. For speech, a rectangular window of typically 20 to 30 ms is often used. The pitch period is measured from the short time autocorrelation function using the procedure described above. The window is advanced, usually by 10 ms, and the analysis repeated. In this way the pitch trajectory is computed pitch asynchronously (cf. Section 7.2.3). Note that each pitch period estimate is the average value over the window, and that pitch period markers are not produced using this approach. Note also that the dynamic range of the pitch period peaks in the autocorrelation function is

usually less than 10 dB, which is considerably less than the (typically) 30 dB dynamic range of peaks in the acoustic signal (Markel, 1972b).

Since the secondary variable t of the short-time autocorrelation function arises from the integration limits of equation (7.6), it will be omitted in the following discussion. The computational steps required to evaluate equations (7.4) (for s(t) of finite duration) and (7.6) are identical. Thus the designation "short time" can often be omitted without introducing complications.

7.4.2 Computational Considerations

Evaluation of equation (7.6) is difficult to perform using analogue techniques, principally because of the necessity for s(t) to be delayed. Until recently, analogue memory technology relied extensively on electromechanical storage devices such as acoustic delay lines and magnetic tape. An extensive review of analogue correlation techniques is given by Lange (1967). The development of digital memory overcame many of the practical problems encountered with analogue correlation, and most of the techniques discussed herein are digitally oriented. Recent research into CCD and SAW technologies may result in a resurgence of interest in analogue processing, especially for very high frequency signals where digital technology is expensive (Heighway, 1976). CCD analogue shift registers employing a "bucket brigade" approach have also been used to construct correlators which operate in the audio range (Fannin, 1975). Since these require the signal to be sampled, a discrete-time representation of the signal is

used and many of the comments pertaining to digital computation apply.

In discrete-time form, the autocorrelation function is given by

$$\phi(\tau) = \sum_{n=1}^{N} s_n \, s_{n+m} \qquad \text{for } m = 0, 1, 2 \ldots N-1 \qquad (7.7)$$

where

$$\tau = m \, \Delta_t \qquad (7.8)$$

and Δ_t is the sampling interval. s_n is the n^{th} sample of the signal, $s(t)$, which is represented by the N sample sequence $[s_1, s_2 \ldots s_N]$, and is zero for n outside the range $1 \leq n \leq N$. Some authors (e.g. Ross, Shaffer, Cohen, Freudberg and Manley, 1974) normalise equation (7.7) by dividing the summation by N. Since s_n is in general non-zero only for $1 \leq n \leq N$, equation (7.7) is more efficiently computed as

$$\phi(\tau) = \sum_{n=1}^{N-|m|} s_n \, s_{n+|m|} \quad . \qquad (7.9)$$

In addition, if the pitch period T is known *a priori* to lie in the range $T_{min} \leq T \leq T_{max}$, then equation (7.9) need be evaluated only for $T_{min} \leq \tau \leq T_{max}$. Further computational efficiency is gained by making use of the fact that many of the s_n appear twice as multiplicands, as Blankinship (1974) has pointed out.

The autocorrelation function is the inverse Fourier transform of the power spectrum, $G(\omega)$, of $s(t)$ (cf. Papoulis, 1962; Lathi, 1965). Thus

$$\phi(\tau) \leftrightarrow G(\omega) = |S(\omega)|^2 \qquad (7.10)$$

where

$$s(t) \leftrightarrow S(\omega) \quad . \tag{7.11}$$

The FFT provides an efficient computational frequency-domain method for evaluating equation (7.7). This is especially true when N is large, or when $\phi(\tau)$ is required for many values of τ (i.e. when the pitch range is large). The evaluation of $\phi(\tau)$ using this approach is performed as follows. The N samples of s(t) are appended with N zeros to form an $\hat{N} = 2N$ length data frame $\{\hat{s}_n\}$. This is required because of the cyclic convolution property of the Fourier transform, which treats the data frame as one period of a periodic signal. The omission of the zeros results in the evaluation of equation (7.7) using the 2N samples $[s_1, s_2, \ldots s_N, S_1, s_2, \ldots s_N]$. The spectrum $\{\hat{S}_n\}$ of $\{\hat{s}_n\}$ is computed using the FFT. Note that many FFT implementations require that N be an integer power of 2 (cf. Bergland, 1969) although this restriction is relaxed to include other values of N when the mixed radix FFT is used (Bergland, 1969; Singleton, 1969). From $\{\hat{S}_n\}$ the power spectrum $\{|\hat{S}_n|^2\}$ is formed. The inverse FFT of the power spectrum results in $\{\hat{\phi}_m\} = \{\phi(\tau)\}$ for $\tau = 0, 1, 2, \ldots N-1$. Since $\{\hat{s}_n\}$ is real the FFT computation may be simplified, resulting in half the computation steps required if $\{\hat{s}_n\}$ were complex (Bergland, 1969).

Further discussion of the use of the FFT method to compute correlations is given in Section 8.2. Other recently developed fast transforms which possess a convolution property analogous to that of the DFT are also considered in Chapter 8.

The number N of signal samples required to evaluate $\phi(\tau)$ for pitch estimation depends strongly upon the expected range of pitch values. This is because N depends upon both the signal sampling rate and the signal window length. The signal sampling rate is governed by the period measurement resolution required. Since the worst case resolution for a given sampling rate occurs when the period is shortest, the sampling rate required to achieve a specified period measurement resolution depends upon the highest pitch value. In contrast, the signal window length depends upon the lowest pitch to be measured, because the window should span two or more periods (Dubnowski, Schafer and Rabiner, 1976). The dependence of N upon the pitch range and measurement resolution is now considered quantitatively.

Denote by f_{max} the frequency corresponding to the highest pitch to be measured. Denote by f_r the required resolution in H_z, and by $f_s = 1/\Delta_t$ the signal sampling rate. Figure 7.3(a) illustrates a signal (i) with pitch at f_{max} and a second signal (ii) which differs from f_{max} by f_r. It is apparent that for the period measurements of signals (a) and (b) to be distinguishable, the sampling frequency f_s must satisfy the condition

$$\frac{1}{f_s} \leq \frac{1}{f_{max} + f_r} - \frac{1}{f_{max}} \quad . \tag{7.12}$$

Rearrangement of equation (7.12) results in

$$f_s \geq f_{max}^2 / f_r + f_{max} \tag{7.13}$$

which shows that if f_r is fixed then f_s depends approximately upon the squre of f_{max} if f_{max} is large. Fortunately, in most applications which require the measurement of pitch values spanning a wide range, the ratio f_r/f_{max} is a more appropriate index of performance than f_r. Denote the ratio f_r/f_{max} by β. For most situations β is typically in the range 0.01 to 0.03, since a ratio of 5.9% corresponds to about a semitone resolution. Then the condition (7.13) becomes

$$f_s \geq f_{max} (1 + 1/\beta) . \tag{7.14}$$

For $\beta = 0.01$ this requires that $f_s \approx 100\ f_{max}$, so that a sampling rate of 40 kHz is adequate for most applications.

Now consider the value of N in terms of the lowest pitched signal. Denote by f_{min} the frequency corresponding to this lowest pitch. Then

$$t_w = \gamma / f_{min} \tag{7.15}$$

is the duration of the window, which spans γ periods of the signal at f_{min}. Typically, $\gamma \geq 2$ as noted above. Now

$$N = f_s\ t_w \tag{7.16}$$

where N is the number of samples in the window. Consequently,

$$N \geq \frac{f_{max}}{f_{min}} \gamma (1 + 1/\beta) . \tag{7.17}$$

At $\beta = 0.05$, $\gamma = 2$, $f_{min} = 40$ Hz and $f_{max} = 3$ kHz this corresponds to a value of N = 3150. This is an order of

magnitude larger than the number of samples used in most speech pitch analysis systems which are based on auto-correlation analysis.

The preceding discussion assumes that the signal sampling rate, once chosen, is held constant, and that the measurement resolution is selected as a worst case value. Once these parameters are fixed, the actual measurement resolution improves as the measured pitch decreases. This suggests that the effective sampling rate which is used should be adaptively altered to achieve the desired resolution. Thus, if $\hat{f}$ is the frequency of the expected pitch value, then the condition (7.17) can be replaced by

$$N \geq \frac{\hat{f} + \Delta}{\hat{f} - \Delta} \gamma (1 + 1/\hat{\beta}) \qquad (7.18)$$

where Δ is the expected variation in $\hat{f}$, and $\hat{\beta} = f_r/\hat{f}$. This requires prior knowledge of the pitch which is to be measured - for example using predictive pitch tracking. Further discussion of this is given in Section 7.4.6.

Another approach which reduces the number N of samples required to compute $\phi(\tau)$ is the technique of "downsampling". This method consists of reducing the signal sampling frequency by a factor which is typically in the range 2 to 5, so that N is also reduced by the same factor (cf. equation 7.16). The discrete autocorrelation function is evaluated and interpolated to form a continuous function. This interpolation permits the peak near $\phi(T)$ to be located with resolution better than a sample interval Δ_t, thus compensating for the loss of resolution incurred by the reduction in sampling rate. Interpolation methods which

have been used include band-limited trigonometric interpolation (Markel, 1972b) and low order spline interpolation (Wise, Caprio and Parks, 1976).

7.4.3 Signal Preprocessing Techniques for Autocorrelation

Signal preprocessing is used in autocorrelation methods of pitch period measurement for either of two reasons. First, the signal can be modified in such a manner that the implementation of the autocorrelator is simplified. Second, formant structure in the signal may be significantly reduced, resulting in a higher correlation between the wave-shapes of adjacent periods. Recently, a real-time hardware system was built by Dubnowski, Schafer and Rabiner (1976) incorporating both these techniques. This is described later in this section.

Typical of preprocessing functions in the first category is hard limiting or infinite peak clipping. This is defined in Table 7.1(a), and results in a binary signal for which multiplication modulo 2 is the EXCLUSIVE OR function (denoted by XOR). Consequently, the expensive multiplier required for the autocorrelator may be replaced by an inexpensive XOR gate. It is thus practicable to build a parallel array of (modulo 2) multipliers for high speed autocorrelation. An additional advantage is that the digital memory requirement is reduced, since only 1 bit per sample is required.

Gill (1961) describes a pitch estimator which employs four single-bit autocorrelators, each operating in parallel on a different prefiltered band of the speech signal.

He comments that while this system works well in many instances, there are also many signals for which it performs poorly. A similar conclusion is reached by Stone and White (1963), who describe a single-bit autocorrelator which uses the wideband speech signal as input.

The reason for the comparatively high failure rate of such systems is not hard to see. Hard limiting preserves only zero crossing and polarity information, and destroys peak and energy distribution information. Experiments by Licklider and Pollack (1948) demonstrate that hard limiting essentially preserves the formant structure of speech. Consequently, the rapid changes in formant structure which are common in speech cause poor performance in the single-bit autocorrelator. In addition, steady-state signals with significant energy at a pitch harmonic will cause an erroneous autocorrelator output corresponding to that harmonic. This situation is common in music, and occurs in speech when interaction exists between the glottal excitation and a vocal tract formant.

The remainder of this section deals with signal preprocessing techniques which reduce formant structure. Such techniques are called "spectrum flattening" (Sondhi, 1968) because the resulting waveform approximates a train of impulses of the same periodicity as the original signal. Sondhi proposes two methods for performing spectral flattening - a filter bank system and centre clipping. Atal (1968) uses cubing to achieve a similar result, although this introduces an undesirable increase in the signal dynamic range.

The filter bank method is a frequency-domain approach to harmonic equalisation. The signal is filtered by a bank of bandpass filters which span the signal bandwidth. The output of each filter is normalised to unit amplitude by dividing it by its short-time energy. The total spectrally-flattened signal is obtained by adding the individually flattened channels with appropriate delays. An additional refinement of this method is phase synchronisation using minimum phase compensation.

As Dubnowski *et al.* (1976) have pointed out, there are several drawbacks to practical implementation of the filter bank method. Considerable hardware is required for filtering and equalisation. Also, while the method works well in many cases, poor results are obtained for signals which have no pitch harmonic occurring within the range of an individual bandpass filter. In this case the filter output level is low, and the equalised output is essentially high level noise which tends to obscure rather than enhance the pitch estimation process.

Sondhi's second method - centre clipping - is much more successful because it does not rely on the presence of a large number of harmonics in the signal. Centre clipping is defined in Table 7.1(b) and was first used in speech by Licklider (1946), who showed that formant structure in speech is significantly destroyed by even a few per cent of centre clipping.

The effect of hard limiting, cubing and centre clipping on the spectrum and autocorrelation function is shown in Figures 7.4 to 7.17 for a variety of instruments

and for voice. Further discussion is given in Section 7.5.6, where the effects of preprocessing are compared for both autocorrelation and cepstral analysis. Rabiner (1977) also presents detailed results of a formal evaluation of the preprocessors defined in Table 7.1 (b), (c) and (d) for autocorrelation pitch analysis of speech, and concludes that all three preprocessors yield similar results.

Dubnowski, Schafer and Rabiner (1976) describe a real-time hardware speech pitch estimator which uses both centre and peak clipping (see Table 7.1(d)). The input analogue speech signal is low pass filtered to a bandwidth of about 900 Hz, and digitally encoded in 12 bit samples at a 10 kHz rate. This digital signal is sectioned into overlapping 30 ms intervals for processing - since the period computation is performed every 10 ms, adjacent sections overlap by 20 ms. For each section, the clipping level is determined by scanning the first and third 10 ms of the section. The maximum absolute signal values in these two intervals are measured. A fixed percentage of the smaller of the two maxima is used as the clipping level for the section. This relatively sophisticated adaptive clipping threshold calculation permits a high clipping level of typically 80% to be used, even when severe envelope changes occur. In contrast, Sondhi uses a smaller level - typically 30% of the peak absolute signal values of the entire signal section.

When the clipping level is determined, the signal section is both centre and peak clipped, so that

$$
\begin{aligned}
x_n &= +1 \quad \text{if} \quad +C_L < s_n \\
&= 0 \quad \text{if} \quad -C_L \leq s_n \leq +C_L \\
&= -1 \quad \text{if} \quad s_n < -C_L
\end{aligned}
\tag{7.19}
$$

where x_n is the n^{th} preprocessor output, C_L is the centre clipping threshold and s_n is the n^{th} signal sample. This preprocessor characteristic is illustrated in Table 7.1(d).

The resulting ternary signal is used as input to the autocorrelator. This is implemented as a pair of fast access bipolar memories (35 ns cycle time), with a "combinatorial multiplier", accumulator and control logic. Since the product required to evaluate the autocorrelation function (equation 7.7) is of the form $x_n\, x_{n+m}$ where x_n has permissible values of +1, 0 or -1, the product is given by

$$
\begin{aligned}
x_n\, x_{n+m} &= 0 \quad \text{if} \quad x_n = 0 \text{ or } x_{n+m} = 0 \\
&= 1 \quad \text{if} \quad x_n = x_{n+m} = \pm 1 \\
&= -1 \quad \text{if} \quad x_n = -x_{n+m} = \pm 1 \ .
\end{aligned}
\tag{7.20}
$$

Thus the individual product terms can be evaluated by a simple combinatorial circuit, and these terms can be summed using an up-down counter.

The autocorrelation function is computed for all values of delay between controllable initial and final values m_{min} and m_{max}. Typical values used are 25 and 200, which correspond to a pitch range of 400 Hz to 50 Hz at the 10 kHz sampling rate. In addition, $\phi(0)$ is computed to permit normalisation of the autocorrelation function.

The maximum value of $\phi(\tau)$, $\tau_{min} \leq \tau \leq \tau_{max}$ is measured, and the corresponding delay $\tau = T \neq 0$ is the pitch period. The voiced-unvoiced decision is made from the ratio $\phi(T) / \phi(0)$, using a threshold of about 0.3 . In addition, a silence threshold is computed from a 50 ms measurement of background noise. If the peak signal value in any 30 ms signal section is below this silence threshold a "silence" condition is indicated, and the autocorrelation evaluation is suppressed. The silence threshold can be reset manually or under program control, so that the detector can adapt to a variety of background environments.

This pitch estimator has been extensively evaluated by Rabiner, Cheng, Rosenberg and McGonegal (1976). Dubnowski *et al.* summarise its performance, and state that most measurements are within the resolution of the estimator. Gross errors occur occasionally (typically once per second) and can be eliminated using nonlinear smoothing (Rabiner, Sambur and Schmidt, 1975). The worst performance occurs for low-pitch speakers, where the pitch period is longer than half the (fixed) analysis frame size. Under these conditions any autocorrelation estimator is expected to perform poorly.

7.4.4 Autocorrelation and Parameter Estimation

Atal (1968) derives a normalised autocorrelation function for pitch synchronous pitch period analysis, by considering period measurement as a parameter estimation problem. He generates a model of an idealised speech signal (the "matching signal") for which the period is uniquely defined. This matching signal is optimised to the actual

signal, and its period used as the pitch period estimate. Observing that speech signals are often similar between adjacent periods, apart from a scale factor which is due to a multiplicative envelope, Atal constrains his matching signal x(t) to the form

$$x(t) = \alpha\, x(t - \tau_1) \tag{7.21}$$

in the interval $t_2 \leq t < t_2 + \min(\tau_1, \tau_2)$, where α is a positive constant and τ_1 and τ_2 are the durations of the first and second pitch periods of x(t). Note that τ_1 and τ_2 are not constrained to be equal, so that inter-period differences are explicitly included in the analysis. Denote the beginning of the first, second and third pitch periods by t_1, t_2 and t_3, and the original signal by s(t). Figure 7.3(b) illustrates these definitions.

The squared error integral

$$I(\tau_1, \tau_2, \alpha) = \int_{t_1}^{t_3} [s(t) - x(t; \tau_1, \tau_2, \alpha)]^2 \, dt \tag{7.22}$$

is a useful indication of how well x(t) approximates s(t). This integral has three degrees of freedom. As outlined below, minimixing $I(\tau_1, \tau_2, \alpha)$ with respect to τ_2 and α yields an error function $I_o(\tau_1)$ dependent only on τ_1. The value of τ_1 for which $I_o(\tau_1)$ is minimised is used as the pitch period value for the first period. The analysis then proceeds pitch synchronously.

Atal's analysis is summarised below. Splitting equation (7.22) into the intervals $[t_1, \min(\tau_1, \tau_2)]$, $[t_1 + \min(\tau_1, \tau_2), t_2]$ and $[t_2, t_3]$, and applying upper

and lower constraints

$$\tau_1' \leq \tau_1 \leq \tau_1''$$

$$\tau_2' \leq \tau_2 \leq \tau_2'' \qquad (7.23)$$

to τ_1 and τ_2 yields an optimal value of $x(t; \tau_1, \tau_2, \alpha)$ given by

$$\hat{x}(t; \tau_1, \tau_2, \alpha) = \frac{1}{1+\alpha^2} [s(t) + \alpha\, s(t + \tau_1)] \qquad (7.24)$$

in the interval $[t_1, t_1 + \min(\tau_1, \tau_2)]$. Back substitution in (7.22) yields

$$I(\tau_1, \tau_2, \alpha) = \frac{1}{1+\alpha^2} \int_{t_1}^{t_1 + \min(\tau_1, \tau_2)} [\alpha\, s(t) - s(t + \tau_1)]^2 \, dt . \qquad (7.25)$$

Since the integrand is positive, the optimum value of τ_2 is τ_2' from equation (7.23). Expanding equation (7.25) and evaluating

$$\frac{\partial I}{\partial \alpha} (\tau_1, \alpha) = 0 \qquad (7.26)$$

yields an optimum value of α given by

$$\hat{\alpha} = \frac{E_2 - E_1}{2 E_{12}} + \sqrt{\frac{(E_2 - E_1)^2}{4 E_{12}^2} + 1} \qquad (7.27)$$

for the case $E_{12} > 0$. E_1, E_2 and E_{12} are defined by

$$E_1 = \int_{t_1}^{t_1 + \theta} s^2(t)\, dt$$

$$E_2 = \int_{t_1}^{t_1 + \theta} s^2(t + \tau_1)\, dt$$

$$E_{12} = \int_{t_1}^{t_1 + \theta} s(t)\, s(t + \tau_1)\, dt \,, \qquad (7.28)$$

where

$$\theta = \min(\tau_1, \tau_2') \,. \qquad (7.29)$$

For the cases $E_{12} = 0$ and $E_{12} < 0$, equation (7.18) yields

$$\hat{\alpha} = 0 \qquad (7.30)$$

which indicates that a meaningful measure of pitch is obtained only for the case $E_{12} > 0$. Since E_{12} is the short-time autocorrelation function (cf. equation 7.6), this result is expected. Substituting equation (7.27) into (7.25) yields

$$I(\tau_1) = \frac{E_1 + E_2 - \sqrt{(E_2 - E_1)^2 + 4\, E_{12}^2}}{2} \qquad (7.31)$$

for $E_{12} > 0$. Normalising $I(\tau_1)$ such that

$$0 \leq I_o(\tau_1) \leq 1 \qquad (7.32)$$

yields

$$I_o(\tau_1) = \frac{(E_1 + E_2) - \sqrt{(E_2 - E_1)^2 + 4\, E_{12}^2}}{2 \min(E_1, E_2)} \qquad (7.33)$$

Forming $J(\tau_1) = 1 - I_o(\tau_1)$ yields

$$J(\tau_1) = \frac{-|E_2 - E_1| + \sqrt{(E_2 - E_1)^2 + 4\,E_{12}^2}}{2\,\min(E_1, E_2)} \quad . \quad (7.34)$$

Note that $J(\tau)$ is computationally more efficient to generate than $I_o(\tau)$, and that $J(\tau)$ is constrained to the interval $[0, 1]$. Also, when $E_1 = E_2$, $J(\tau)$ reduces to E_{12} / E_1, which corresponds to the autocorrelation function normalised by the signal energy.

Atal's approach is an interesting extension of direct short-time autocorrelation, since it explicitly considers adjacent period differences in both amplitude scale and in period. However the calculation of $J(\tau)$ from equation (7.34) is computationally more expensive than evaluation of the autocorrelation function. In addition, the frequency domain transformation method discussed in Section 7.4.2 does not apply. To the author's knowledge this method has not been compared with other standard techniques such as auto-correlation or cepstral analysis. The fact that Atal's method has not been followed up with subsequent evaluation suggests that the trade-off between performance and complexity is not promising. In fact Atal (1972) comments that simpler techniques are preferable, at least for pitch asynchronous analysis, and mentions the Gold and Rabiner (1969) parallel processing method as a suitable alternative.

For completeness, it should be noted here that Atal introduced signal cubing as a spectral flattening preprocessing technique (Section 7.4.3). This preprocessing is incorporated into his analysis system, and its effect on

his published results should not be overlooked.

Pitch estimation has also been considered as a parameter estimation problem by Wise, Caprio and Parks (1976), who note that Noll (1970) independently proposed a similar approach. Since Wise *et al.* introduce modifications which overcome some of the problems encountered by Noll, the latter's method is not discussed here.

The maximum likelihood method of Wise *et al.* is formulated as the problem of estimating an unknown periodic signal in white Gaussian noise of unknown intensity. This problem has been considered also by Caprio (1975), and is summarised below. Let s_k be a periodic repetition of the P length sequence q_k, so that

$$s_k = q_{k \bmod P} . \tag{7.35}$$

The measured signal r_k of length K consists of s_k and additive noise n_k so that

$$r_k = s_k + n_k \qquad \text{for } k = 1, 2, \ldots N \tag{7.36}$$

or in vector notation

$$\underset{\sim}{R} = \underset{\sim}{S} + \underset{\sim}{N} . \tag{7.37}$$

The n_k are independent Gaussian random variables with zero mean and variance σ^2. From $\underset{\sim}{R}$ it is required to estimate the signal $[q_1, q_2, \ldots q_p]$, its period P and the noise intensity σ^2.

By considering $n_k = r_k - s_k$ to be a k-dimensional Gaussian random vector, the conditional probability density

function of the measured signal is formulated as

$$p(\underset{\sim}{R}/P,\ \underset{\sim}{Q},\ \sigma^2) = \frac{1}{(2\pi\sigma^2)^{K/2}} \exp\left[-\frac{1}{2\sigma^2}\sum_{k=0}^{K-1}(r_k - s_k)^2\right] \quad (7.38)$$

The maximum likelihood estimator derived below maximises equation (7.38), or equivalent maximises

$$\ln p(\underset{\sim}{R}/P,\ \underset{\sim}{Q},\ \sigma^2) = -\frac{K}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{k=0}^{K-1}(r_k - s_k)^2 . \quad (7.39)$$

Recognising the periodic nature of $\underset{\sim}{S}$, the sum in the second term of equation (7.39) is rewritten as

$$\sum_{\ell=0}^{N-1}\sum_{k=0}^{P-1}(r_{k+\ell P} - q_k)^2 + \sum_{k=0}^{K-NP-1}(r_{k+NP} - q_k)^2 \quad (7.40)$$

where N is defined as the greatest integer less than $[K/P]$. Note that when K is an integer multiple of P the second term of equation (7.40) is zero.

Thus

$$\frac{\partial \ln}{\partial q_k} p(\underset{\sim}{R}/P, \underset{\sim}{Q}, \sigma^2) = \begin{cases} \frac{1}{\sigma^2}\sum_{\ell=0}^{N}(r_{k+\ell P} - q_k), & k < K-NP \\ \frac{1}{\sigma^2}\sum_{\ell=0}^{N-1}(r_{k+\ell P} - q_k), & k \geq K-NP . \end{cases} \quad (7.41)$$

Equating this to zero yields

$$\hat{q}_k(P) = \begin{cases} \frac{1}{N+1}\sum_{\ell=0}^{N} r_{k+\ell P}, & k < K - NP \\ \frac{1}{N}\sum_{\ell=0}^{N-1} r_{k+\ell P}, & k \geq K - NP . \end{cases} \quad (7.42)$$

Similarly, equating

$$\frac{\partial \ln\, p(\underset{\sim}{R}/P,\ \underset{\sim}{Q},\ \sigma^2)}{\partial \sigma^2} \;=\; 0 \tag{7.43}$$

gives

$$\hat{\sigma}^2(P) \;=\; \frac{1}{K} \sum_{k=0}^{K-1} (r_k - \hat{s}_k)^2 \tag{7.44}$$

where

$$\hat{s}_k \;=\; \hat{q}_{k \bmod P} \,.$$

It now remains to choose $\hat{P}$ as that value of P which maximises $\ln p(\underset{\sim}{R}/P,\ \underset{\sim}{Q},\ \sigma^2)$. Combining equations (7.42) and (7.44) with (7.39) shows that the maximum is achieved by minimising

$$\sum_{k=0}^{K} (r_k - \hat{s}_k)^2$$

or equivalently by minimising $\hat{\sigma}^2(P)$.

Combining equation (7.42) with (7.44) gives

$$\hat{\sigma}^2(P) \;=\; \frac{1}{K} \left[\sum_{k=0}^{K-1} r_k^2 - \sum_{k=0}^{K-1} \hat{s}_k^2 \right]$$

$$\;=\; \frac{1}{K} \left[\phi_R(o) \;-\; E_{\hat{s}}(P) \right] \tag{7.45}$$

where

$$\phi_R(k) \;=\; \sum_{j=0}^{K-1-k} r_j\, r_{j+k} \tag{7.46}$$

is the autocorrelation function of $\underset{\sim}{R}$, and

$$E_{\hat{s}}(P) \;=\; \sum_{k=0}^{K-1} \hat{s}_k^2 \;=\; (N+1) \sum_{k=0}^{K-NP-1} \hat{q}_k^2 \;+\; N \sum_{k=K-NP}^{P-1} \hat{q}_k^2 \tag{7.47}$$

is the energy of $\hat{s}$. Since $\hat{P}$ is that value of P which maximises $E_{\hat{s}}(P)$, the problem is reduced to a one dimensional maximisation.

In the case where the signal measurement spans an integer number of periods, i.e. K = NP, equation (7.47) reduces to

$$E_{\hat{s}}(P) = N \sum_{k=0}^{P-1} \hat{q}_k^2 \quad . \tag{7.48}$$

Applying equations (7.42) and (7.46) to (7.48) gives

$$E_{\hat{s}}(P) = \frac{P}{K} \left[\phi_R(0) + 2 \sum_{\ell=1}^{N-1} \phi_R(\ell P) \right] \quad . \tag{7.49}$$

Thus for the special case where the signal spans an integer number of periods, $E_{\hat{s}}(P)$ can be expressed either in terms of the signal estimate $\hat{q}_k(P)$ (equation 7.47) or in terms of the autocorrelation function $\phi_R(k)$ of the observed signal (equation 7.49). If N is sufficiently large, the value of $E_{\hat{s}}(P)$ given by equation (7.49) is approximately the same as that given by (7.47), even for values of P which are not integer submultiples of K. This makes the formulation of equation (7.49) useful for pitch estimation. Thus the pitch period estimate is that value of P which maximises $E_{\hat{s}}(P)$, or equivalently minimises $\hat{\sigma}^2(P)$.

Wise *et al.* modify this formulation using a result of Caprio (1975), to reduce the effect of noise on the peaks of $E_{\hat{s}}(P)$, and to introduce the equivalent of the linear taper in the short time autocorrelation function (Section 7.4.1). This results in the function

$$g(P) \triangleq \frac{2P}{K} \sum_{\ell=1}^{N-1} \phi_R(\ell p)$$

$$= \hat{E_s}(P) - \frac{P}{K} \phi_R(0) \quad . \tag{7.50}$$

The value of P at which g(P) is maximised is the period estimate. Wise *et al.* present results for speech, and note that g(P) can be interpolated to obtain period resolution finer than one sampling period. They also give an interpretation of their method as a comb filter which passes the maximum signal energy. This contrasts with the methods discussed in Section 7.4.5, which null the periodic component of the signal.

7.4.5 Optimum Comb Filter

The comb filter (Moorer, 1974) is defined by the recurrence relation

$$y_n = s_n - s_{n-m} \tag{7.51}$$

where s_n is the input signal sample at time $n\Delta_t$, y_n is the output signal sample at time $n\Delta_t$, m is a constant integer which defines the characteristics of the filter, and Δ_t is the sampling interval. The magnitude-frequency response of the comb filter is

$$\{\sin^2(m\omega\Delta_t) + [1 - \cos(m\omega\Delta_t)]^2\}^{\frac{1}{2}} \quad , \tag{7.52}$$

so that a zero of transmission occurs at the frequencies which are integer multiples of $1/m\Delta_t$ Hz. Consequently if the input is a stationary signal consisting only of frequencies which are integer multiples of $1/m\Delta_t$ Hz

then the steady state filter output is zero. This suggests that periodicity can be estimated by determining the parameter m of the comb filter which results in the minimum filter output. Since the speech and music signals of interest are nonstationary, the optimum comb filter is evaluated using successive windowed signal segments.

The optimum comb filter is determined by minimising

$$\sum_{n=1}^{N} (s_n - s_{n-m})^2 \tag{7.53}$$

with respect to m, over the N length windowed signal. Since no simple minimisation technique is applicable other than trial and error (Moorer, 1974), the function

$$\sum_{n=1}^{N} |s_n - s_{n-m}| \tag{7.54}$$

is also a suitable indication of optimality, with a computational advantage over (7.53). Denote by $\hat{m}$ the value of m which minimises (7.54). Then the pitch period estimate T of the N sample signal sequence $[s_1, s_2, \ldots s_n]$ is given by

$$T = \hat{m}\Delta_t \ . \tag{7.55}$$

Note that for a periodic signal the minimum is not unique, because any integer multiple of m, including zero, will also produce a minimum in (7.54). In addition, since s_n is zero outside the range $1 \leq n \leq N$, the summation (7.54) contains fewer non-zero terms as m is increased. This results in the equivalent of the linear taper which is

discussed in Section 7.4.1. However, while this linear taper is advantageous for autocorrelation (since period multiples are weighted unfavourably with respect to the period), the converse is true for the function (7.54). Moorer does not mention this taper problem, but he apparently recognises it because he evaluates (7.54) over precisely one period. While his explanation is not explicit, it appears that he evaluates

$$\sum_{n=1}^{M} |s_n - s_{n-m}| \tag{7.56}$$

for $T_{min} \leq m\Delta_t \leq T_{max}$, where for speech applications T_{min} and T_{max} correspond to 225 Hz and 70 Hz respectively. By choosing a window length $N\Delta_t$ which is nearly $2T_{max}$, the taper effect is virtually eliminated since few terms in the summation (7.56) are zero.

Ross, Shaffer, Cohen, Freudberg and Manley (1974) introduce the function (7.54) as a degenerate form of autocorrelation, rather than as a comb filter. They define the average magnitude difference function as

$$D(\tau) = \frac{1}{N} \sum_{n=0}^{N-1} |s_n - s_{n-m}| , \quad m = 0, 1, \ldots m_{max} \tag{7.57}$$

where $\tau = m\Delta_t$, and Δ_t and s_n have their usual meanings. To overcome the linear taper effect, Ross *et al.* constrain the summation in equation (7.57) to the range $m \leq n \leq N-1$, so that $D(\tau)$ is formed only in the region of overlap of the sequences s_n and s_{n-m}.

Ross *et al.* derive an approximate relationship between $D(\tau)$ and $\phi(\tau)$. From Schwarz's inequality (cf. Taub and

Schilling, 1971), the bound

$$\frac{1}{N}\sum_{n=0}^{N-1}|X_n| \leq \left[\frac{1}{N}\sum_{n=0}^{N-1}X_n^2\right]^{\frac{1}{2}} \qquad (7.58)$$

is established. This permits $D(\tau)$ to be approximated as

$$D(\tau) = \frac{1}{N}\sum_{n}|s_n - s_{n-m}| \cong \beta(\tau)\left[\frac{1}{N}\sum_{n}(s_n - s_{n-m})^2\right]^{\frac{1}{2}} \qquad (7.59)$$

where $\beta(\tau)$ is a scale factor which depends on the joint probability density function of s_n and s_{n-m}. Ross *et al.* note that $\beta(\tau)$ is typically in the range 0.6 to 1.0, and does not depend strongly on τ. Defining the auto-correlation function as

$$R(\tau) = \frac{1}{N}\sum_{n} s_n s_{n-m} \qquad (7.60)$$

(cf. equation 7.7), expanding equation (7.59), and assuming that $\{s_n\}$ is stationary yields

$$D(\tau) \cong \beta(\tau)\,[2\,(R(0) - R(\tau)]^{\frac{1}{2}} \quad . \qquad (7.61)$$

Ross *et al.* observe that the approximation (7.61) is a reliable characterisation of $D(\tau)$.

Since $D(\tau)$ (equation 7.57) is the same as the comb filter optimisation function (equation 7.54) apart from a scale factor, both methods are equivalent. $D(\tau)$ is computationally more attractive than $\phi(\tau)$ if the latter is computed in the time domain (cf. Section 7.4.2) because subtracters are usually faster and/or cheaper than multipliers. Ross *et al.* discuss the implementation of a real-time pitch estimator using a GTE Sylvania programmable

signal processor, which has instruction execution times of typically 250 to 375 ns. They conclude that for speech sampled at about 7 kHz, pitch estimates in the range 70 Hz to 300 Hz can be computed in about 10 msec. This period computation rate is sufficiently fast for real time operation. However, as Rabiner, Cheng, Rosenberg and McGonegal (1976) note, the computation time depends approximately on the square of the sampling rate, so that this method is not attractive for the wide pitch range encountered with music signals. It is doubtful whether downsampling (cf. Section 7.4.2) would alleviate this problem.

Miller and Weibel (1956) describe analogue instrumentation which incorporates a delay line memory and which evaluates $D(\tau)$ (equation 7.57) for periodicity measurement.

7.4.6 Heuristics for Time Domain Computation

The computation required to estimate the pitch period T from $\phi(\tau)$ (or the related functions discussed above) can be reduced by employing pitch tracking heuristics (Moorer, 1974; Rabiner, Cheng, Rosenberg and McGonegal, 1976; Wise, Caprio and Parks, 1976). This approach assumes that adjacent period estimates are similar, so that $\phi(\tau)$ need be evaluated only in the neighbourhood of $\phi(T_{j-1})$, where T_{j-1} denotes the previous period estimate. The initial period estimate is obtained from a global search of $\phi(\tau)$. Since this pitch tracking can lead to the tracking of a secondary maximum (cf. Figure 7.2(e), which illustrates a pitch doubling signal transition) it is desirable to perform a

global search of $\phi(\tau)$ at regular intervals. Moorer (1974) performs this global search once every 100 ms and employs a back-up procedure if erroneous pitch tracking is detected. This global search and back-up procedure is also invoked if two successive period estimates differ by more than a predetermined tolerance. Note that the back-up procedure which corrects erroneous pitch tracking is suitable for "off-line" pitch processing, but is not possible for real-time operation. In the latter situation a more appropriate approach is to adaptively alter the interval between adjacent global searches.

These pitch tracking heuristics reduce the computation required to determine T from $\phi(\tau)$ only if $\phi(\tau)$ is evaluated using the time-domain method, since the frequency-domain method results in the global evaluation of $\phi(\tau)$. As noted in Section 7.4.2, frequency-domain computation of $\phi(\tau)$ is more efficient than time-domain computation if $\phi(\tau)$ is required for many values of τ or if the signal frame contains many samples. Consequently pitch tracking heuristics are unlikely to be useful if the pitch range extends over 5 or 6 octaves.

7.5 FREQUENCY DOMAIN METHODS

In this section are discussed several pitch measurement methods which require transformation of the acoustic signal into the frequency domain.

7.5.1 Product Spectrum

As noted in Section 7.3 the fundamental frequency

component of a periodic signal is often contaminated with noise or is missing completely. This situation is especially common for telephone speech signals. Miller (1953) found that the fundamental frequency of a speech signal can be accurately derived by dividing the measured frequencies of the second, third or fourth harmonics by 2, 3 or 4 respectively. A practical difficulty with this approach is that the harmonic numbers are not always known reliably.

Harris and Weiss (1963) refine this method by measuring the frequency difference between widely separated harmonics, which occur at frequencies f_j and f_{j+n-1}. The number n-2 of spectral peaks which correspond to intermediate harmonics is also measured. The pitch frequency is then computed as

$$f_o = (f_{j+n-1} - f_j) / (n-1) . \qquad (7.62)$$

However, this method requires extensive heuristics to minimise the risk of selecting spectral peaks which correspond to formant frequencies rather than to pitch harmonics, and to ensure that all intermediate pitch harmonic peaks are counted.

Observations by Flanagan and Golden (1966) indicate that for speech the instantaneous frequencies measured at the outputs of narrow-band filters differ little from integer multiples of the fundamental for frequencies up to at least 2500 Hz. Thus the fundamental frequency can be estimated from these instantaneous frequencies (which correspond to the pitch harmonic frequencies) by dividing

each measured frequency by the corresponding harmonic number. This proposal also suffers from the lack of *a priori* knowledge of the harmonic numbers. Schroeder (1968) suggests that this difficulty is overcome by assuming that the harmonic numbers of the measured frequencies are relatively prime (i.e. that their largest common divider is unity). Thus, all integer submultiples of the measured frequencies are computed, and that submultiple which is most common is the fundamental frequency estimate. For example, if the measured filter output frequencies are 777 Hz and 1000 Hz, then the fundamental estimate is 111 Hz and the measured frequencies correspond to the 7th and 9th harmonics, respectively.

Schroeder describes analogue instrumentation which incorporates a bank of band-pass filters to construct in real time a frequency histogram (or alternatively the corresponding period histogram). From this the pitch trajectory of speech is computed in real time. Schroeder found that better results are obtained if the contribution of each harmonic to the histogram is weighted by the logarithm of its spectral amplitude. In a computer simulation of this method he obtained speech pitch period measurements identical to those produced using cepstral analysis (cf. Section 7.5.3). Since this approach requires only a single spectral transformation, unlike cepstral analysis which requires a double transformation, it is computationally superior.

Schroeder generalises the weighted frequency histogram by applying it to a continuous rather than

discrete spectrum. Using Schroeder's notation:

$$\Sigma(f) \triangleq 20 \log_{10} \sum_{n=1}^{N} |S(nf)| \qquad (7.63)$$

where $S(f)$ is the short time spectrum of the acoustic signal $s(t)$, and n and N are integers. The antilogarithm of $\Sigma(f)$ is called the "harmonic product spectrum" of $s(t)$. Schroeder notes that $\Sigma(f)$ is related to the cepstrum $c(\tau)$ (Section 7.5.3) by

$$\sigma(\tau) = \prod_{n=1}^{N} c(\tau / n) \quad , \qquad (7.64)$$

where $\sigma(\tau)$ is the Fourier cosine transform of $\Sigma(f)$. He also notes that Noll, in his efforts to refine cepstral analysis, has introduced functions similar to the product spectrum (cf. Noll, 1970). The logarithmic product spectrum of speech is suitable for direct pitch measurement since it usually possesses a sharp absolute maximum at the fundamental frequency. This is true even when significant additive Gaussian noise is present. Under these conditions the corresponding peak in the cepstrum is frequently obscure.

It is interesting to note that despite the computational and claimed performance superiority of the product spectrum over the cepstrum, the former has received little subsequent attention, while the cepstrum is often used as a comparison standard when evaluating new pitch measurement methods (Markel, 1972b; Moorer, 1974).

Since the product spectrum and frequency histogram rely on the presence of pitch harmonics, they are not

suitable for pitch analysis of some musical instruments (e.g. flute, french horn). This is discussed further in Section 7.5.6.

7.5.2 Hilbert Transform

If $s(t)$ is a real band-limited signal, the corresponding analytic signal $\psi(t)$ is defined as

$$\begin{aligned} \psi(t) &= s(t) + j\,\hat{s}(t) \\ &= \alpha(t)\,\exp[j\,\theta(t)] \end{aligned} \tag{7.65}$$

where $\hat{s}(t)$ is the Hilbert transform of $s(t)$, and $\alpha(t)$ and $\theta(t)$ are the amplitude and phase respectively of $\psi(t)$ (cf. Papoulis, 1962; Westman, 1968). The application of the Hilbert transform to analysis of modulated signals is well known (cf. Westman, 1968) - for example $\alpha(t)$ and $\theta(t)$ are identical to the envelope and phase, respectively, of a single-sideband modulated signal. Other applications are discussed by Gold, Oppenheim and Rader (1970), in connection with digital signal processing. It is worth noting here that the bandwidth of speech can be reduced by a factor of about 2 using the technique of "analytic signal rooting", in which the signal $\mathrm{Re}[\psi^{\frac{1}{2}}(t)]$ is formed (Schroeder, Flanagan and Lundry, 1967). In addition, Robson (1976) uses $\alpha(t)$ and $\theta(t)$ to identify orchestral musical instruments.

Lerner (1959) and Rader (1964) have constructed pitch estimators which use Hilbert transform techniques. Since Rader's method is essentially an extension of Lerner's method, only the former is discussed here.

Rader calls his method "vector pitch detection".

The input signal is applied to a bank of N band-pass filters. The output of each band-pass filter is passed through a 90° phase-shifting circuit (which approximates a Hilbert transformer). The N filter outputs and the N phase shifted outputs constitute the components of a vector in 2N space. When the input signal is periodic, the vector traces a closed curve in 2N space, and the time taken to traverse this curve is equal to the period. If several of the band-pass filters contain significant energy then the curve is unlikely to be self-intersecting, so that the vector will return to its "starting point" only once every pitch period. Moreover, even if the curve is self-intersecting, the vector will return to its "starting point" too early only if the "starting point" and the point of curve self-intersection are coincident. This is very unlikely if the "starting point" is selected randomly.

Rader implements his method by generating a "distance measure" D(t) between the instantaneous vector position and the "starting point". The latter is defined by sampling the vector components at an instant specified by the pitch estimator control circuitry. D(t) is formed by squaring the difference between the instantaneous and starting point values of each vector component, and summing these squared values.

When the input signal is only quasi-periodic, D(t) has a local minimum after one pitch period. Rader suggests that improved performance can be achieved by employing pitch tracking (cf. Section 7.4.6). He also observes that for speech the local minimum in D(t) at the pitch period T is

also usually the global minimum over the interval $0.3T \leq t \leq 1.8T$.

Sondhi (1964) asserts that the vector pitch estimator is essentially an autocorrelation pitch estimator. Sondhi shows that if narrow band filters are used, then

$$D(\tau) \cong 2\pi [\phi(0) - \phi(\tau)] \tag{7.66}$$

where $\phi(\tau)$ is the short time autocorrelation function, and τ is the time interval which has elapsed since the vector starting point was sampled. However, Bluestein and Rader (1965) point out that Sondhi's analysis is applicable only if an infinite number of infinitesimally narrow band-pass filters are used in the vector pitch estimator. They provide a detailed comparison of both vector and autocorrelation pitch estimation, and show that for practical filter banks there exist substantial differences between the two methods.

7.5.3 Cepstrum

The cepstrum was introduced by Bogert, Healy and Tukey (1963) in an attempt to estimate the time delay between a seismic signal and its echoes. Cepstral analysis has since been applied to a variety of problems in seismology and geophysics (Cohen, 1970; Ulrych, 1971; Hassab, 1974), in bioelectric data processing (Kemerait and Childers, 1972), in image deblurring and image classification (Rom, 1975) and in speech analysis-synthesis (Oppenheim, 1969), as well as to speech pitch estimation (Noll, 1964, 1967; Markel, 1972b; Rabiner, Cheng, Rosenberg and McGonegal, 1976). A feature common to many of these

problems is that two or more signals are combined by convolution rather than by addition. The de-convolving property possessed by the cepstrum can aid the separation of the constituent signals - a technique called homomorphic filtering (Oppenheim, Schafer and Stockham, 1968). The properties of the cepstrum and some of its relatives are now discussed, and its application to pitch estimation of speech and other signals is described.

Consider a composite signal $s(t)$ which is the convolution of two component signals. As noted in Section 7.3, speech can be modelled as such a composite signal, where the glottal excitation $g(t)$ and the vocal tract impulse response $v(t)$ are the convolved components. Then

$$s(t) = g(t) * v(t) \tag{7.67}$$

where the operator * denotes convolution. Denote by $S(\omega)$, $G(\omega)$ and $V(\omega)$ the Fourier transforms of $s(t)$, $g(t)$ and $v(t)$ respectively, and assume that these transforms exist. Then

$$S(\omega) = G(\omega) \cdot V(\omega) \quad . \tag{7.68}$$

For voiced speech $g(t)$ can be modelled as a quasi-periodic train of impulses with period T. Consequently $s(t)$ is also quasi-periodic with period T, and the power spectrum $|S(\omega)|^2$ contains harmonics spaced at $1/T$ Hz. These harmonics are manifested as periodic "ripples" in the power spectrum (see Figure 7.4(b)). The period of these ripples can be measured by taking the Fourier transform of

the power spectrum. The peak which corresponds to the "frequency" of the power spectrum ripples occurs at 1/(1/T), or T sec. This process is essentially that of auto-correlation analysis (cf. Section 7.4.2) and is illustrated in Figure 7.4(a). Thus,

$$\begin{aligned}\phi(\tau) &\equiv \mathcal{F}\,[|S(\omega)|^2] \\ &= \mathcal{F}\,[|G(\omega)|^2 \cdot |V(\omega)|^2] \\ &= \mathcal{F}\,[|G(\omega)|^2] * \mathcal{F}\,[|V(\omega)^2|] \\ &= \phi_g(\tau) * \phi_v(\tau) \end{aligned} \qquad (7.69)$$

where $\mathcal{F}[a]$ denotes the Fourier transform of a, and $\phi_g(\tau)$ and $\phi_v(\tau)$ denote the autocorrelation functions of $g(t)$ and $v(t)$ respectively. The effects of the glottal excitation and vocal tract are thus convolved with each other in the autocorrelation function of speech. This disadvantage of autocorrelation pitch analysis in speech is noted by Schroeder (1966, 1970). Various signal preprocessing techniques which reduce the formant structure (and hence the effect of $\phi_v(t)$) are discussed in Section 7.4.3. Figure 7.4(a) illustrates the effect of the vocal tract response on the autocorrelation function, while Figure 7.6(a) shows the kind of improvement which is obtained when the formant structure is suppressed by centre-clipping the speech signal.

Now consider the logarithm of the power spectrum.

$$\begin{aligned}\log|S(\omega)|^2 &= \log[|G(\omega)|^2 \cdot |V(\omega)|^2] \\ &= \log|G(\omega)|^2 + \log|V(\omega)|^2 \quad . \end{aligned} \qquad (7.70)$$

The cepstrum $c(\tau)$ is defined as

$$c(\tau) \triangleq \mathcal{F}[\log|S(\omega)|^2]$$

$$= \mathcal{F}[\log|G(\omega)|^2] + \mathcal{F}[\log|V(\omega)|^2] \qquad (7.71)$$

(Markel, 1972b). The excitation and response signals are now added rather than convolved. Thus if $G(\omega)$ and $V(\omega)$ occupy separate frequency regions they may be easily separated by "filtering" the logarithm power spectrum. Following Tukey's terminology (Noll, 1967), the term "quefrency" (with units of seconds) is used to describe the "frequency" of the ripples in the logarithm power spectrum. The effect of the vocal tract is to produce a low quefrency ripple in the logarithm power spectrum, while the periodicity of the glottal source is manifested as a high quefrency ripple in the logarithm power spectrum. Therefore, the cepstrum has a sharp peak corresponding to the high quefrency glottal source $g(t)$ and a broader peak corresponding to the low quefrency vocal tract response $v(t)$. Figure 7.4(b) illustrates the spectrum, power spectrum and logarithm power spectrum, while Figure 7.4(a) shows the corresponding cepstrum and identifies the peak which occurs at quefrency T. For most speech the main contribution of $v(t)$ to the cepstrum is contained in the first two or three milliseconds quefrency. Thus the vocal tract response is virtually eliminated by setting to zero the first two or three milliseconds of the cepstrum (Noll, 1967; Markel, 1972b).

The application of cepstral analysis to pitch estimation is described in detail by Noll (1964, 1967). Weiss, Vogel and Harris (1966) describe a real-time hardware

cepstrum pitch estimator. This implementation contains two analogue spectrum analysers and a wide-band analogue delay line memory. Digital evaluation of the cepstrum using the FFT is described by Schafer and Rabiner (1970) and by Markel (1972b). An evaluation and comparison of the cepstrum with other pitch estimation methods is given by Rabiner, Cheng, Rosenberg and McGonegal (1976).

It is worth commenting that Noll (1964) defines the cepstrum as $c^2(\tau)$, where $c(\tau)$ is the cepstrum defined here by equation (7.71). This squared cepstrum produces a sharper peak at quefrency T, and results in a more attractive representation since low amplitude peaks are suppressed. However, as Markel (1972b) points out, the use of $c^2(\tau)$ rather than $c(\tau)$ does not simplify the location of the cepstral peak corresponding to the pitch period, because the dynamic range of this cepstral peak is also squared.

Noll (1967) considers analytically the effect of windowing the signal $s(t)$ to obtain a short-time cepstrum (cf. Section 7.4.1). This analysis shows that the time window $w(t)$ which multiplies $s(t)$ should be tapered smoothly, so that its spectrum $W(\omega)$ has a narrow main lobe and low-amplitude side lobes. The well-known Hamming and Hanning windows are often used (Noll, 1967; Oppenheim, 1969).

Hassab and Boucher (1976) consider the effect of the signal-to-noise ratio on the usefulness of the cepstrum in estimating echo delays. Additive Gaussian noise is assumed, and the mean reduction in the cepstrum peak at delay T is

evaluated as a function of the input total signal-to-noise ratio and the relative signal-to-noise bandwidths.

7.5.4 Clipstrum

Noll (1968) describes a variation of the cepstrum which he calls the clipstrum. The clipstrum is defined by

$$\text{clipstrum } (\tau) = \mathcal{F} [\text{sgn } [\log|F(\omega)|^2]] \qquad (7.72)$$

where

$$F(\omega) = \mathcal{F} [f(t)]$$

and f(t) is a signal derived from s(t) by suitable preprocessing such as hard limiting or centre clipping (cf. Section 7.4.3). In practice it is desirable to eliminate the vocal tract contribution to the logarithm power spectrum (cf. Section 7.5.3) before hard limiting. This is achieved by high-pass filtering (or equivalently by low-quefrency liftering) the logarithm power spectrum - an operation which Noll includes in his definition but which is omitted here for clarity.

The development of the clipstrum was motivated by the desire to simplify the arithmetic required to evaluate the cepstrum. The usefulness of hard-limiting to achieve such arithmetic simplicity is discussed in Section 7.5.3. If the signal preprocessing used is hard limiting, so that f(t) = sgn[s(t)], then the clipstrum performs poorly as a pitch estimator. The reason for this is that the formant structure of s(t) is enhanced rather than suppressed in f(t) (cf. Section 7.5.3). However, if centre-clipping is used then the formant structure of s(t) is suppressed in f(t)

and the clipstrum performs well. Noll does not make it clear how much of the improvement is due to the initial centre-clipping and how much to the hard limiting of the logarithm power spectrum. He does, however, state that the latter procedure introduces additional pitch harmonics to the logarithm power spectrum, consequently enhancing the clipstrum peak at quefrency T. It is worth noting that the author has independently proposed centre-clipping as a useful preprocessing technique for cepstral analysis of signals which possess few harmonics (e.g. the flute). This is discussed further in Section 7.5.6, where results are presented.

7.5.5 Hapstrum

The computational advantage of the fast Walsh transform over the fast Fourier transform (cf. Section 8.4.1) led to the development of the Hapstrum (Tanaka, 1972). The hapstrum is defined as

$$\text{hapstrum}\ (\tau) \quad = \quad \mathcal{W}[\log|S(\omega)|^2] \tag{7.73}$$

where $\mathcal{W}[a]$ denotes the Walsh (or Hadamard) transform of a, and $S(\omega)$ is the Fourier transform of $s(t)$. Tanaka reports that the hapstrum is inferior to the cepstrum for pitch estimation because the peak at quefrency T is not as clearly defined in the hapstrum as in the cepstrum. This conclusion is consistent with the comments made in Sections 8.4 and 8.5, concerning the use of the Walsh transform. It should be noted here that Tanaka also investigates the application of the hapstrum to formant-tracking, using an approach similar to that of Schafer and Rabiner (1970).

However, while the use of the hapstrum rather than cepstrum reduces the computation time required, the accuracy of the estimated formant frequencies is also degraded.

7.5.6 Comparison of Autocorrelation and Cepstrum

It is apparent from the discussion in Section 7.5.3 that the cepstrum performs poorly as a pitch estimator for signals which can not be modelled by equation (7.67). Since most musical instrument signals are of this kind (cf. Section 7.3) cepstral analysis is not a generally-applicable method. It is also worth pointing out that the cepstrum also performs poorly for those speech signals which contain few pitch harmonics, as Rabiner, Cheng, Rosenberg and McGonegal (1976) demonstrate. However in the latter case the use of a preprocessor which introduces additional pitch harmonics should cause improvement (see the paragraph which contains equations (7.70) and (7.71)).

The results of a comparison between autocorrelation and cepstral pitch analysis for a selection of musical instruments and speech is presented in Figures 7.8 to 7.17. These results confirm the comments of the preceding paragraph, and demonstrate that autocorrelation analysis is significantly more successful than cepstral analysis for the signals produced by musical instruments. Figures 7.8 to 7.17 also show the kind of improvement which is achieved for both autocorrelation and cepstral analysis by the use of preprocessing functions (cf. Table 7.1). The effect of these on the signal spectrum, power spectrum and logarithm power spectrum is illustrated in Figures 7.4 to 7.7.

Examination of Figures 7.8 to 7.17 shows that centre clipping is the most universally successful preprocessing function for both autocorrelation and cepstral pitch analysis. While cubing will in general improve both the autocorrelation and cepstrum pitch estimators, this improvement is not universal, as can be seen by comparing parts (a) and (b) of Figure 7.11. For those waveforms which are characterised by possessing only two zero-crossings per pitch period, infinite peak clipping causes more improvement than cubing (e.g. Figures 7.9 to 7.11). However this is not true for signals which exhibit resonances (e.g. Figure 7.14 - see also the fourth paragraph of Section 7.4.3).

7.6 LINEAR PREDICTION AND INVERSE FILTERING METHODS

A speech analysis-synthesis method called linear prediction is formulated in the time-domain by Atal and Schroeder (1970) and Atal and Hanauer (1971). This method and its variants have subsequently received considerable attention, both for speech analysis (cf. Makhoul, 1973; Maksym, 1973; Crichton and Fallside, 1974; Itakura, 1975; Knudsen, 1975) and synthesis (cf. Haskew, Kelley, Kelley and McKinney, 1973; Sambur, 1975; Rabiner and Schafer, 1976), as well as for diverse applications such as recovery of "helium speech" (Atal and Hanauer, 1971) and speech encryption (Sambur and Jayant, 1976). Markel (1972a) points out that linear prediction is a special case of the method formulated by R. Prony in 1795, which was extended to a least-squares formulation in 1924 by C. Runge and H. Konig. Makhoul (1975) presents an extensive and detailed

review of linear pediction, and discusses its applications to neurophysics and geophysics as well as to speech. Makhoul also points out that the all-zero linear predictor corresponds to the well known "moving average" model used in statistics, while the all-pole linear predictor and pole-zero linear predictor correspond, respectively, to the "autoregressive" model and "autoregressive moving average" model (cf. Box and Jenkins, 1970).

This section presents a brief review of linear prediction and inverse filtering techniques for speech analysis, with emphasis on pitch estimation. For a comprehensive review and extensive bibliography, the reader is referred to Makhoul (1975). Section 7.6.1 outlines the time-domain linear prediction formulation of Atal and Hanauer (1971), which Makhoul (1973) calls the covariance method. In Section 7.6.2 the inverse filtering method of Markel (1972a, 1972b) is discussed - this approach is also called the autocorrelation method of linear prediction (Makhoul, 1973). It should be pointed out here that these techniques are essentially equivalent, and are related also to the maximum likelihood estimator which is derived in terms of the autocorrelation function in Section 7.4.4 (cf. Markel, 1972a; Makhoul, 1973). Makhoul (1975) also points out that the inverse filter formulation is identical to the method of maximum entropy spectral estimation (cf. Burg, 1972; Van den Bos, 1971). Provided these points are remembered, the terminology used herein should cause no confusion.

7.6.1 Linear Prediction

The essential feature of linear prediction as applied to speech analysis and synthesis is that speech is assumed to be generated as the output of an all-pole filter which is excited once every pitch period (for voiced speech) or by white noise (for unvoiced speech). The all-pole model can be extended to include zeros (cf. Kopec, Oppenheim and Tribolet, 1977) although this latter model is not considered here. Atal and Hanauer (1971) point out that for non-nasalised voiced speech the vocal tract transfer function has no zeros. Also, for speech which is unvoiced or nasal, all the zeros of the vocal tract transfer function lie within the unit circle of the z plane and can therefore be approximated by multiple poles, so that the all-pole model of the vocal tract is adequate. Atal and Hanauer also show that the effect of the glottal volume flow and the radiation can be accounted for by a two pole transfer function, which can thus be incorporated with the vocal tract transfer function into a single all-pole filter.

The all-pole filter (with the number of poles denoted by p) is formulated digitally as a recursive weighted sum of the previous p samples of the filter input, plus the excitation signal (hence the terminology "linear prediction"). Denote by s_n the speech signal sample at time $n\Delta_t$, where $\Delta_t = 1/f_s$ is the sampling interval. Then the linear predictor output is

$$s_n = \sum_{k=1}^{p} a_k s_{n-k} + \delta_n \qquad (7.74)$$

where the predictor coefficients a_k account for the filtering action of the vocal tract, the glottal volume flow, and the radiation; and δ_n denotes the n^{th} sample of the excitation. The transfer function of the filter is

$$T(z) = 1 / (1 - \sum_{k=1}^{p} a_k z^{-k}) \qquad (7.75)$$

so that the filter has p poles which must be either real or complex conjugate pairs. Additionally, for the filter to be stable, all the poles must lie inside the unit circle of the z domain.

Atal and Hanauer show analytically that for speech the number of poles required for an adequate representation of the vocal tract transfer function can be estimated from the requirement that the duration of the linear predictor memory be twice the time taken for sound to propagate from the glottis to the lips (or, for nasal sounds, to the nasal opening). For a typical adult whose vocal tract is approximately 17 cm long, the predictor memory duration should be approximately 1 ms. If the signal sampling rate f_s is 10 kHz, the corresponding number of poles is 10. Thus, p should be approximately 12 (taking into account the two additional poles which are required to account for glottal volume flow and radiation). This result is confirmed experimentally. It should however be remembered that p depends strongly upon f_s.

A complete description of the speech signal over a time interval during which the vocal tract shape is assumed to be constant is provided by the predictor coefficients a_k (which are estimated from the speech signal in the manner

described below), the pitch period, the rms value of the speech samples, and a binary parameter which indicates the nature of the excitation (i.e. voiced or unvoiced). The continual variations in the vocal tract shape which are encountered in practice can be adequately accounted for by periodic readjustment of these parameters, for example once every 5 or 10 ms.

The method used by Atal and Hanauer to evaluate the predictor coefficients a_k from the speech signal is now described. Define the prediction error e_n as

$$e_n = s_n - \hat{s}_n$$

$$= s_n - \sum_{k=1}^{p} a_k \, s_{n-k} \qquad (7.76)$$

where

$$\hat{s}_n = \sum_{k=1}^{p} a_k \, s_{n-k} \qquad (7.77)$$

is the n^{th} predicted speech sample, and s_n is the n^{th} speech sample. Define the mean-squared prediction error $\langle e_n^2 \rangle_{av}$ as

$$\langle e_n^2 \rangle_{av} = \langle \left(s_n - \sum_{k=1}^{p} a_k \, s_{n-k} \right)^2 \rangle_{av} \, . \qquad (7.78)$$

Observe that this definition omits the effect of the excitation function δ_n. The predictor coefficients a_k are chosen such that $\langle e_n^2 \rangle_{av}$ is minimised, i.e. as the solution of

$$\frac{\partial \langle e_n^2 \rangle_{av}}{\partial a_k} = 0 \, , \qquad k = 1, 2, \ldots p \, . \qquad (7.79)$$

Atal and Hanauer show that the condition (7.79) can be formulated as

$$\sum_{k=1}^{p} \phi_{jk} a_k = \phi_{j0} \quad , \quad j = 1, 2, \ldots p \tag{7.80}$$

where

$$\phi_{jk} = \langle s_{n-j} s_{n-k} \rangle_{av} \quad . \tag{7.81}$$

Equation (7.80) can be written in matrix form as

$$\Phi \underset{\sim}{a} = \underset{\sim}{\psi} \tag{7.82}$$

where $\Phi = [(\phi_{jk})]$ is a positive definite symmetric covariance matrix, and $\underset{\sim}{a} = [(a_k)]$ and $\underset{\sim}{\psi} = [(\phi_{j0})]$ are column vectors. The nature of Φ (viz. positive definite, symmetric) permits equation (7.82) to be solved efficiently (cf. Faddeev and Faddeeva, 1963; Makhoul, 1975). Atal and Hanauer also comment that occasionally the coefficients a_k computed in this manner produce poles in the predictor transfer function which lie outside the unit circle of the z domain. They describe a simple procedure to detect such poles and correct the predictor coefficients.

The prediction error e_n defined by equation (7.76) does not take into account the excitation function δ_n, as has been pointed out. This forms the basis of a pitch period estimation method, since for voiced speech $\{\delta_n\}$ approximates an impulse train (see also Maksym, 1972, 1973, who describes a real-time hardware implementation). Thus, e_n will also approximate an impulse train, and a relatively simple peak-picking procedure is sufficient to estimate each pitch period. However, for other excitation functions (such

as those of musical instruments - cf. Section 7.3) this approach is not satisfactory, as the author has confirmed (Tucker, 1974). Consequently this pitch estimation method is not generally applicable for signals other than speech.

Atal and Hanauer also consider application of the linear predictor to speech synthesis, and provide a recording which illustrates their results. The formulation of equation (7.74) is used directly. The control parameters supplied to the synthesiser are the pitch period, the binary voiced-unvoiced parameter, the rms value of the speech samples, and the p predictor coefficients. A pulse generator produces a pulse of unit amplitude at the beginning of each period, while a white noise generator produces uncorrelated uniformly distributed random samples at each sampling instant. A voiced-unvoiced switch selects the appropriate excitation, whose amplitude is scaled by a factor G which is appropriate to the required rms signal level. The linearly predicted value $\hat{s}_n$ is combined with the excitation signal δ_n to form the n^{th} sample of the synthesised speech. The samples s_n are presented to a DAC, and low-pass filtered to produce the continuous speech signal s(t). The synthesiser control parameters are updated at the start of each pitch period (for voiced speech) or once every 10 ms (for unvoiced speech).

It is worth pointing out that linear-prediction analysis-synthesis provides considerable intrinsic flexibility, which is necessary for many applications such as text-to-speech conversion of connected speech (cf. Rabiner and Schafer, 1976). For example, the speech rate

can be altered without change in pitch, and vice-versa. The formant frequencies and bandwidths can be independently altered, so that a "male voice" can be transformed into a "female voice" and vice-versa. Intonation and stress contours can be produced from stored information about individual words.

It is also worth commenting that the transfer function of an idealised acoustic tube is equivalent to that of the linear prediction model, as Atal and Hanauer show. This property is exploited by Crichton and Fallside (1974), who use linear prediction to compute and display in real time the estimated vocal tract shape. This display forms a useful aid to the teaching of deaf children (see also Section 7.7).

7.6.2 Inverse Filtering

Markel (1972a, 1972b, 1973) presents a frequency-domain formulation of all-pole speech modelling, and discusses its application to formant analysis as well as to pitch estimation. Makhoul (1973, 1975) derives an identical model from a frequency-domain consideration of linear prediction. The latter approach is outlined here, because it relates more obviously to the discussion of Section 7.6.1.

Consider the z transform of the prediction error e_n defined by equation (7.76):

$$E(z) = \left[1 + \sum_{k=1}^{p} a_k z^{-k} \right] S(z)$$

$$= A(z)\, S(z) \qquad (7.83)$$

where the inverse filter $A(z)$ is defined by

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} \qquad (7.84)$$

and where $E(z)$ and $S(z)$ are the z transforms of $\{e_n\}$ and $\{s_n\}$, respectively. Therefore, e_n can be regarded as the result of passing s_n through the inverse filter $A(z)$. Recall that $\{e_n\}$ is assumed to consist of an impulse at the start of each pitch period (for voiced speech) or white noise (for unvoiced speech) (cf. Section 7.6.1), so that the spectrum of $\{e_n\}$ for one pitch period is flat. Thus the inverse filter $A(z)$ can be regarded as a "whitening filter".

The coefficients a_k of $A(z)$ are evaluated by requiring the energy of $\{e_n\}$ to be minimised (cf. equations 7.76 to 7.82), or equivalently be requiring that the average of the magnitude spectrum $|E(z)|^2$ evaluated on the unit circle be minimised. Markel (1972b) shows that $\{a_k\}$ can be obtained as the solution to

$$\sum_{k=1}^{p} a_k r_{k-j} = - r_j \qquad j = 1, 2, \ldots p \qquad (7.85)$$

where

$$r_j = \sum_{n=0}^{N-1-|j|} s_n s_{n+|j|} \qquad (7.86)$$

and the signal $\{s_n\}$ is of length N. Since $\{r_j\}$ is the autocorrelation function of $\{s_n\}$ (cf. equation 7.9) this formulation is called the autocorrelation method of linear prediction.

Markel (1972b) describes a speech pitch estimation algorithm called SIFT (simplified inverse filter tracking) which is more than an order of magnitude faster in operation than cepstral analysis, while yielding results of comparable accuracy (cf. Rabiner, Cheng, Rosenberg and McGonegal, 1976). However the SIFT algorithm is applicable only to speech, for the reasons mentioned in Section 7.6.1. It is also worth pointing out that the SIFT algorithm performs poorly for speech which has few pitch harmonics, as Rabiner *et al.* (1976) demonstrate.

7.7 PITCH FROM GLOTTAL MEASUREMENTS (SPEECH)

The techniques reviewed so far attempt to extract the pitch period from the acoustic signal which, for speech, may be approximated as the convolution of the glottal excitation and vocal tract response functions (cf. Section 7.3). This section reviews speech pitch analysis methods which are based on direct measurement of vocal cord movement. Since the source-filter model of speech production does not apply to musical instruments (Section 7.3), no analogous process exists for music signal analysis.

Signals derived from glottal movement possess considerably simpler waveshapes than the corresponding acoustic speech signal. Consequently, many of the difficulties encountered by the pitch measurement methods reviewed above do not apply, and simple real-time period measurement techniques may be used. Such simple techniques include threshold crossing interval and peak interval measurements.

Despite the simplified pitch estimation methods which are possible using signals derived from glottal movement, comparatively little interest has been shown in this approach. This is presumably because of the need for the speaker to wear sensors which are in physical contact with his throat. It is ironical that this limitation did not discourage many earlier workers from considering the use of throat microphones, which produce a signal which is similar to the acoustic signal and which contains much of the vocal tract response (McKinney, 1965).

Four different techniques are currently available for direct measurement of vocal cord movement - laryngoscopy, trans-glottal illumination, air flow measurement and electro-glottography (Fourcin and Abberton, 1971). Of these, electro-glottography is the only approach suitable for pitch estimation.

7.7.1 Laryngoscopy

Laryngoscopy involves the illumination and viewing of the vocal cords, usually with the aid of an interposed mirror at the back of the mouth, near the naso-pharynx. The laryngeal frequency can be measured using a stroboscope (Flanagan, 1972), or with the high-speed photographic technique developed by Farnsworth (1940). Normal speech is not possible during this procedure.

7.7.2 Trans-glottal Illumination

Trans-glottal illumination uses a small supra-glottal light source which is introduced through one nostril and the naso-pharynx. The light path through the vocal cords is

monitored by a photo-cell pressed externally against the throat. The photo-cell output signal correlates well with the vocal cord displacement (Sonesson, 1960). However it is difficult to maintain the light source in position, and its introduction is unpleasant for many speakers.

7.7.3 Air Flow Measurement

Air flow from the lungs through the vibrating vocal cords has a pulsating component superimposed on an approximately steady velocity. The pulsating component is due to the periodic blocking of the tracheal air stream by the vocal cords, and has the same periodicity as the vocal cord vibration. This pulsating velocity component of the direct air velocity measurement has not been used for laryngeal frequency measurement (Fourcin and Abberton, 1971). However a related signal has been derived by inverse filtering of the pressure (i.e. acoustic) speech waveform (Miller, 1959). More recent inverse filtering methods are discussed in Section 7.6.

7.7.4 Electro-glottography

The electrical impedance between electrodes situated on either side of the throat at the level of the larynx is largely dependent on the position of the laryngeal components (Fourcin, Donovan and Roach, 1971). The dependence on vocal cord separation is of particular interest, and has been exploited to monitor vocal cord movement during normal speech by Fabre (1959) and more recently by Fourcin and Abberton (1971).

7.7.5 The Laryngograph

Of the four methods outlined above, only the electrical impedance monitoring approach is suitable for monitoring larynx vibration during normal speech, as Fourcin *et al.* (1971) have pointed out. Research by Fourcin and others at the Department of Phonetics, University College London, has resulted in the commercial manufacture and distribution of a device - the Laryngograph - which monitors and displays in real time the pitch contours of speech.

The Laryngograph uses a pair of screened electrodes which are placed on either side of the speaker's throat at larynx level. One electrode is excited by a 1 MHz sinusoidal signal. The signal received by the other electrode is processed by a compensation network, which compensates for gross impedance differences between different speakers and for long-term fluctuations due to oesophageal movement and vertical displacement of the larynx. The receiving electrode signal is approximately independent of the glottal aperture, and depends predominantly on the degree of vocal cord contact.

Normal voicing produces a waveform with three distinct parts in each period. A typical waveform is shown in Figure 7.18(a). A sharp rise is produced by the rapid closing of the vocal cords - this corresponds to the interval of greatest vocal tract excitation. Immediately following this is a gentler fall, which is associated with the gradual parting of the vocal cords as the sub-glottal pressure is increased. Between successive glottal closures the waveform is relatively flat, corresponding to the

interval during which the glottis is open and the vocal cords are out of contact. Normal speakers using normal voicing always produce this type of waveform, which is regularly repeated at the laryngeal frequency rate. However people with pathological conditions of the larynx often produce quite different waveshapes, such as those shown in parts (b) and (c) of Figure 7.18. This has led to the suggestion that the laryngograph (displaying the electrode output signal) be used as a medical diagnosis tool.

Because the laryngograph electrode output signal is simple in shape and has one predominant peak in each period during normal voiced speech, simple techniques can be used to measure pitch trajectories in real time. In addition the voiced-unvoiced decision can be easily incorporated because of the absence of significant peaks in the laryngograph signal during unvoiced intervals.

It is worth noting here that the real time display of pitch trajectories has recently been applied to learning situations where visual feedback is a useful complement to, or replacement of, normal aural feedback. The frequency display uses a logarithmic rather than linear scale, to correlate better with subjective pitch perception. Applications where this approach is proving useful include the teaching of foreign languages (Abberton and Fourcin, 1973) and the teaching of deaf children (Abberton, 1972). It should also be pointed out that a similar application of visual feedback to music teaching is described by Lamb (1977) (see also Tucker, Bates, Frykberg *et al.*, 1977).

In this latter application the MOD display (cf. Chapter 3) provides the pitch-trajectory information.

7.8 HEURISTIC METHODS

The pitch estimation techniques reviewed so far attempt to measure periodicity (or quasi-periodicity) in a signal using mathematical techniques. Gold (1962a) observes that pitch estimation is essentially a pattern recognition problem, and can be approached by heuristically characterising the set of "rules" which humans apply when they perform pitch estimation from a visual signal display (cf. Section 7.2.3). In this section several methods which use this heuristic approach are described.

7.8.1 Parallel Processing Methods

The concept of processing in parallel a number of separate "features" of the signal waveform (and then suitably combining the results) arises naturally when the human visual pattern recognition facility is considered. Thus, a human takes into account not only the amplitudes of the signal peaks but also their "shape". This point is discussed further in Chapter 9, where a new pitch estimation algorithm which employs signal feature recognition is presented.

Gold (1962a) describes a pitch estimation method which applies tests for both peakedness and regularity of the waveshapes of the (wide band) speech signal. Gold's method consists of four main stages. First, the voiced and unvoiced speech segments are distinguished. This is done by

determining the positive or negative maxima and positive or negative minima of the signal. The average of the amplitude separations between adjacent minima and maxima is computed over an interval which is greater than the largest expected period (20 ms). If this average is smaller than a preset threshold then the segment is declared "unvoiced".

Second, voiced segments are subjected to three tests, which corresponds to parallel processing by three independent processors. One test compares adjacent maxima and assigns a tag to each maximum which exceeds the previous maximum by a predetermined constant threshold. Another test assigns a tag to all maxima for which the difference between it and the previous minimum exceeds an adaptive threshold (which is defined as 85% of the short-time signal envelope magnitude). The third test is similar to the first except that the threshold used is time-varying - the previous maximum amplitude is decreased linearly with time.

Third, the peaks which have been tagged in this manner are examined for the regularity of their spacing. Those peaks which are tagged by all three tests are examined initially, by considering all such peaks $\{P_i\}$ in groups of five. Let five adjacent peaks $P_1, P_2, \ldots P_5$ occur at times $t_1, t_2, \ldots t_5$ respectively. Then if

$$\frac{(t_{i+1} - t_i) + (t_i - t_{i-1})}{(t_{i+1} - t_i) - (t_i - t_{i-1})} > R_n , \quad i = 2, 3, 4 \qquad (7.87)$$

the peaks $P_1, P_2, \ldots P_5$ are tentatively tagged as "pitch period marker peaks". R_n is a predetermined threshold, and

the subscript n refers to the search "mode" which is discussed below. If the condition (7.87) is satisfied then the search window is advanced and the peaks $P_2, P_3, \ldots P_6$ are examined similarly. However, if condition (7.87) is not satisfied the search "mode" is altered, by considering the peaks which are tagged by two (instead of three) tests. A total of ten search modes are provided for. These correspond to: all three tests satisfied (1 mode), two tests satisfied (3 modes), one test satisfied (3 modes), and one or another test satisfied (3 modes). Should the regularity examination fail for all ten search modes then no peaks in the current voiced segment are tagged as pitch period marker peaks, and the next voiced segment is processed. Failure also occurs if adjacent peaks P_i are spaced too widely.

Finally, the tentative pitch period markers are "edited" to produce the final pitch period markers (from which the pitch trajectory is computed). This stage is required because sometimes legitimate period marker peaks (as obtained by "eye detection") can fail the tests and/or regularity examination. Similarly, the tests sometimes tag a spurious peak. The editing procedure is essentially a pitch trajectory smoothing algorithm. When sudden period irregularities occur the pitch trajectory is "smoothed" and the largest peak near the smoothed value is investigated as a possible period marker peak. In this way tentative marker peaks may be deleted or new marker peaks inserted.

Following the development of Gold's (1962a) method, Gold (1962b) produced a heuristic algorithm which uses six

comparatively simple period estimators, which operate in parallel on the signal maxima and minima, and determine the periodicity of contiguous peaks and valleys both individually and in pairs. The six independent period estimates are then combined using a "majority vote" procedure which takes into account the fact that some estimators may be incorrectly measuring the second or third harmonics. This algorithm was subsequently implemented in hardware (Daggett, 1966). Gold (1964) describes its application to the voiced-unvoiced decision. Gold and Rabiner (1969) describe the algorithm and two simplifying modifications, and present results. A modified version suitable for musical signals with wide pitch trajectory bandwidths is described in Chapter 9, where it is compared with our new algorithm.

7.8.2 Single Feature Methods

Reddy (1967) describes an algorithm which searches for "significant peaks" in the signal waveform, and which assumes that these significant peaks are spaced at approximately uniform intervals. The first step in the algorithm is to locate all local maxima and minima in the current signal segment. The amplitude of the absolute maximum in the segment is measured. The significant maximum and minimum peaks are then located. A significant maximum peak is defined to be a local maximum which satisfies the following requirements:

(i) It is positive.

(ii) It does not occur within 2.5 ms of the previous significant maximum peak.

(iii) It is either (a) greater than 0.9 times the absolute maximum, or (b) greater than the linearly extrapolated value of the previous two significant maximum peaks, or (c) if neither condition (a) nor (b) are satisfied within 13.5 ms from the previous significant maximum, then it is defined as the maximum of all the local maxima in that 13.5 ms interval.

A significant minimum peak is defined similarly.

These significant maximum and minimum peaks are next examined by an editing procedure. It is required that a significant minimum peak exist within a neighbourhood (3.5 ms wide) of each significant maximum peak, and vice versa. In this way the significant maximum and minimum peaks are "matched" in pairs - such a matched combination is designated as a "significant peak marker", and its time of occurrence is defined as the time of occurrence of the corresponding significant maximum. The significant peak markers are subjected to regularity tests. These require that each individual period estimate (as obtained from the significant peak markers) must not differ from the "most likely" pitch or from the preceding period estimate by more than a fixed percentage. The "most likely" pitch is defined to be the mode of the discrete statistical distribution formed by considering the number of pitch periods that fall within each of the 1 ms intervals between 2 ms and 14 ms. The regularity test procedure contains provision for deleting or inserting significant peak markers.

Miller (1975) describes an algorithm which is similar

to Reddy's. Miller approaches the pitch estimation problem by attempting to locate the "principal excursion cycles" of the signal. The pitch period markers are defined as the zero-crossing instants which correspond to the start of each principal excursion cycle. An excursion cycle is defined as the portion of signal between two consecutive zero crossings. The principal excursion cycles play a similar role in Miller's algorithm to that played in Reddy's algorithm by the "significant peaks".

The values of the amplitude and time of occurrence of the largest value of each excursion cycle are stored as "potential principal excursion cycles", provided the following conditions are met:

(i) The square root of the energy of the excursion cycle exceeds a preset threshold (this distinguishes between voiced and unvoiced speech).

(ii) The time separation between consecutive potential excursion cycles exceeds 2 ms.

If condition (ii) is not satisfied then the excursion cycle whose energy is largest is selected and stored. The stored potential principal excursion cycles are then examined to isolate the true principal excursion cycles. Regions of continuous voiced speech are partitioned into syllabic intervals by examining the signal envelope. A piecewise-linear approximation to the envelope during the syllabic interval is constructed. A discrete period histogram is computed, using as pitch period markers all the potential principal excursion cycles whose amplitudes exceed 0.9 of the piecewise-linear envelope approximation.

The period which occurs most frequently is the "most likely pitch period" of the syllable. The potential principal excursion cycles are then examined and edited using a procedure similar to that described by Reddy.

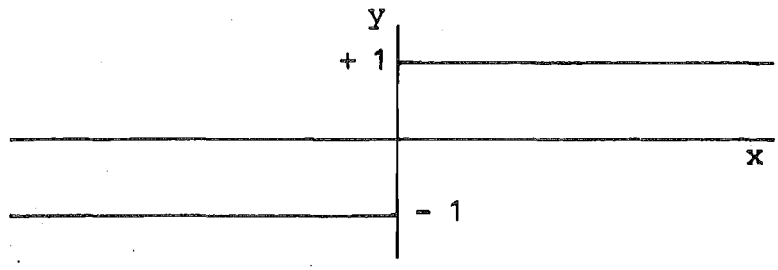
It is worth noting here that formal linguistic analysis has recently been applied to pattern-recognition problems in image processing. Many of these problems are concerned with identifying or matching the "shape" of waveforms of a signal f(x). Consequently, there is common ground between these image processing problems and pitch estimation. Ehrich and Foith (1976) give a useful discussion of such formal linguistic methods. They also consider the representation of the peaks and valleys in a signal using a "relational tree" formalism. To the author's knowledge such techniques have not yet been applied to pitch estimation.

TABLE 7.1

PREPROCESSING FUNCTIONS FOR AUTOCORRELATION AND CEPSTRAL ANALYSIS.

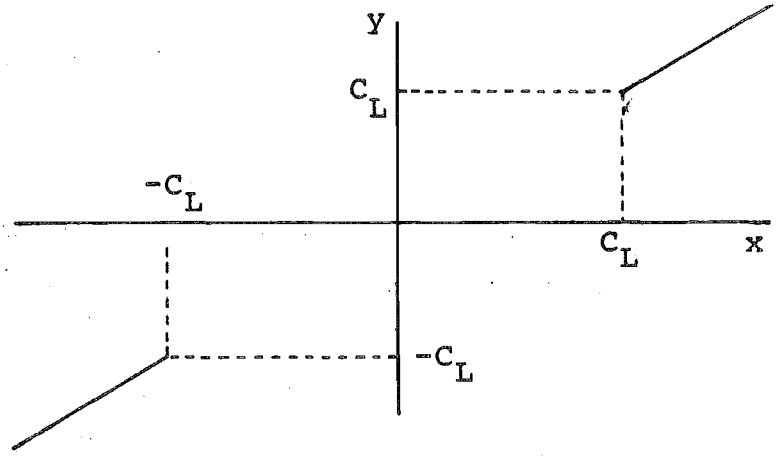
y(t) DENOTES PREPROCESSOR OUTPUT, x(t) DENOTES INPUT.

(a) INFINITE PEAK CLIPPING



$$y(t) = \mathrm{sgn}[x(t)]$$

(b) CENTRE CLIPPING



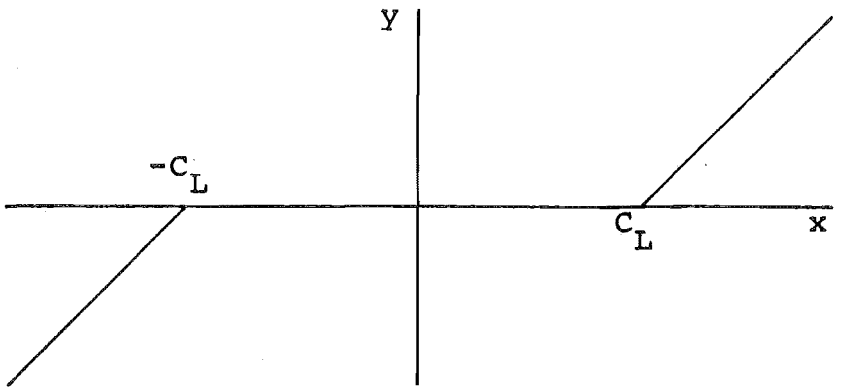
$$y(t) = \mathrm{clp}[x(t)] = x(t), \quad x(t) \geq C_L$$

$$= 0, \quad |x(t)| < C_L$$

$$= x(t), \quad x(t) \leq -C_L$$

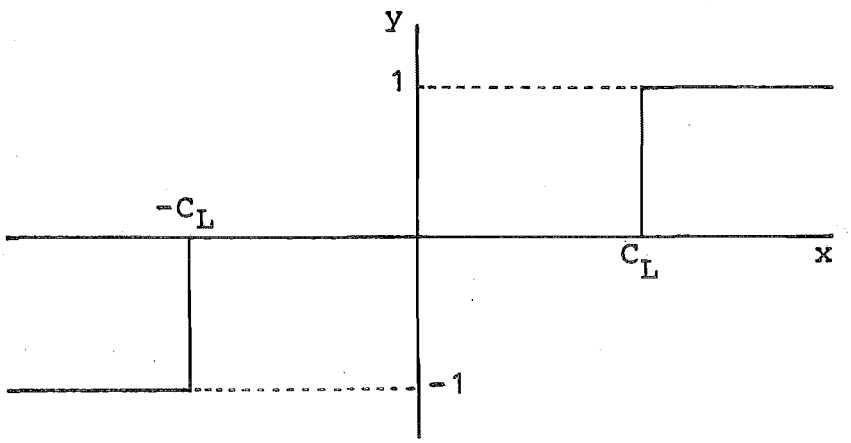
TABLE 7.1 (Continued 2).

(c) COMPRESSED CENTRE CLIPPING



$$y(t) = clc[x(t)] = (x(t) - C_L), \quad x(t) \geq C_L$$
$$= 0, \quad |x(t)| < C_L$$
$$= (x(t) + C_L), \quad x(t) \leq -C_L$$

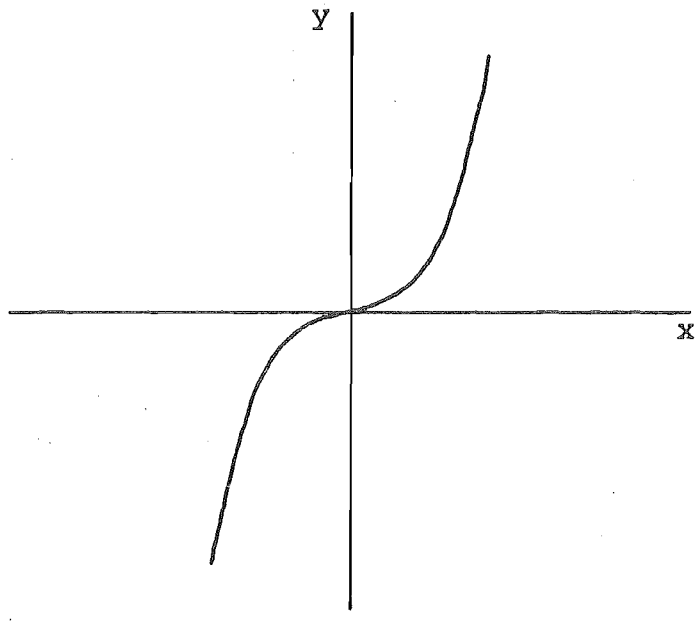
(d) COMBINED CENTRE AND PEAK CLIPPING



$$y(t) = cpc[x(t)] = 1, \quad x(t) \geq C_L$$
$$= 0, \quad |x(t)| < C_L$$
$$= -1, \quad x(t) \leq -C_L$$

TABLE 7.1 (Continued 3).

(e) CUBING



$$y(t) = x^3(t)$$

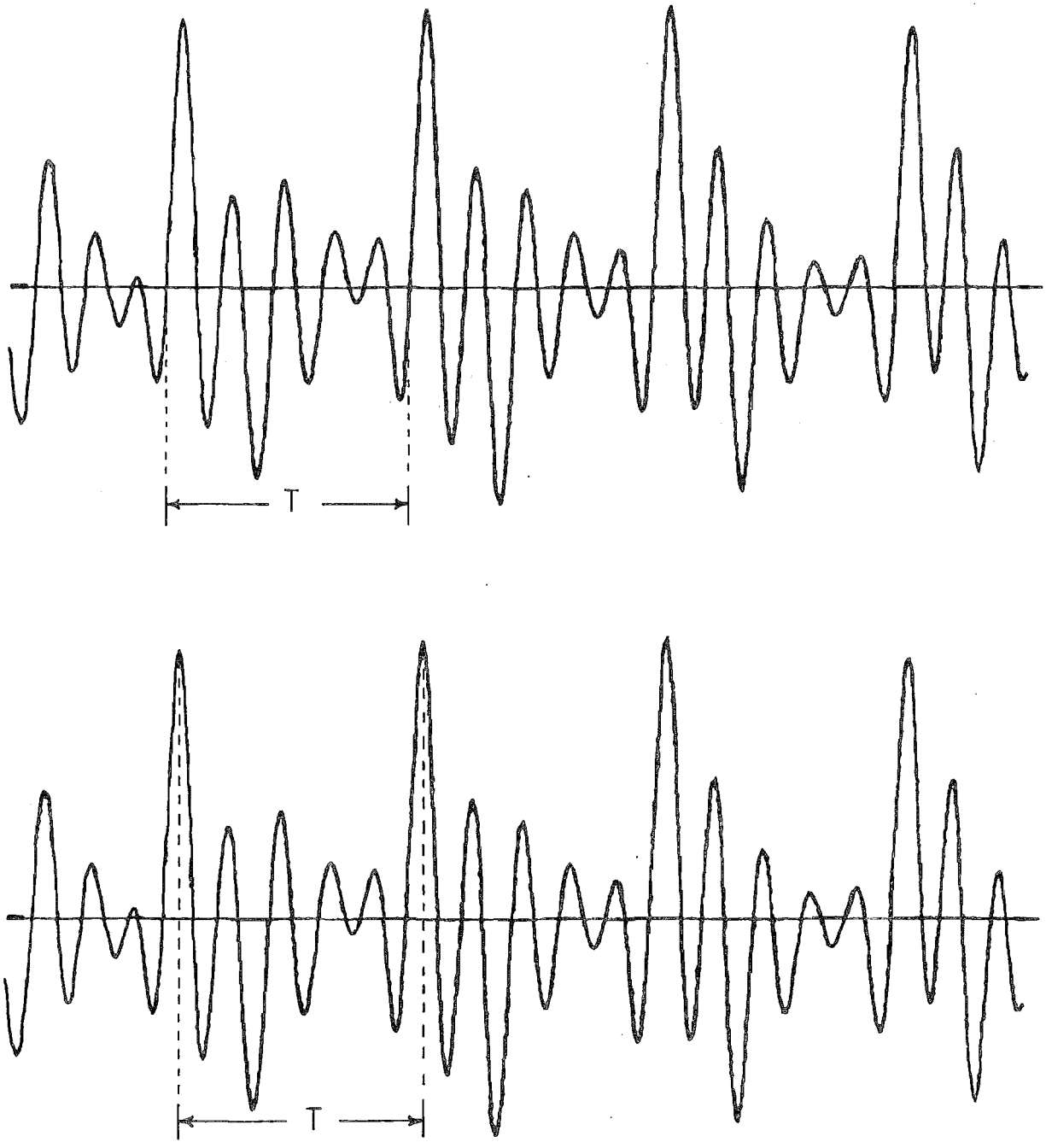


Figure 7.1 Pitch period markers:

(a) coincident with zero crossing

(b) coincident with principal peak.

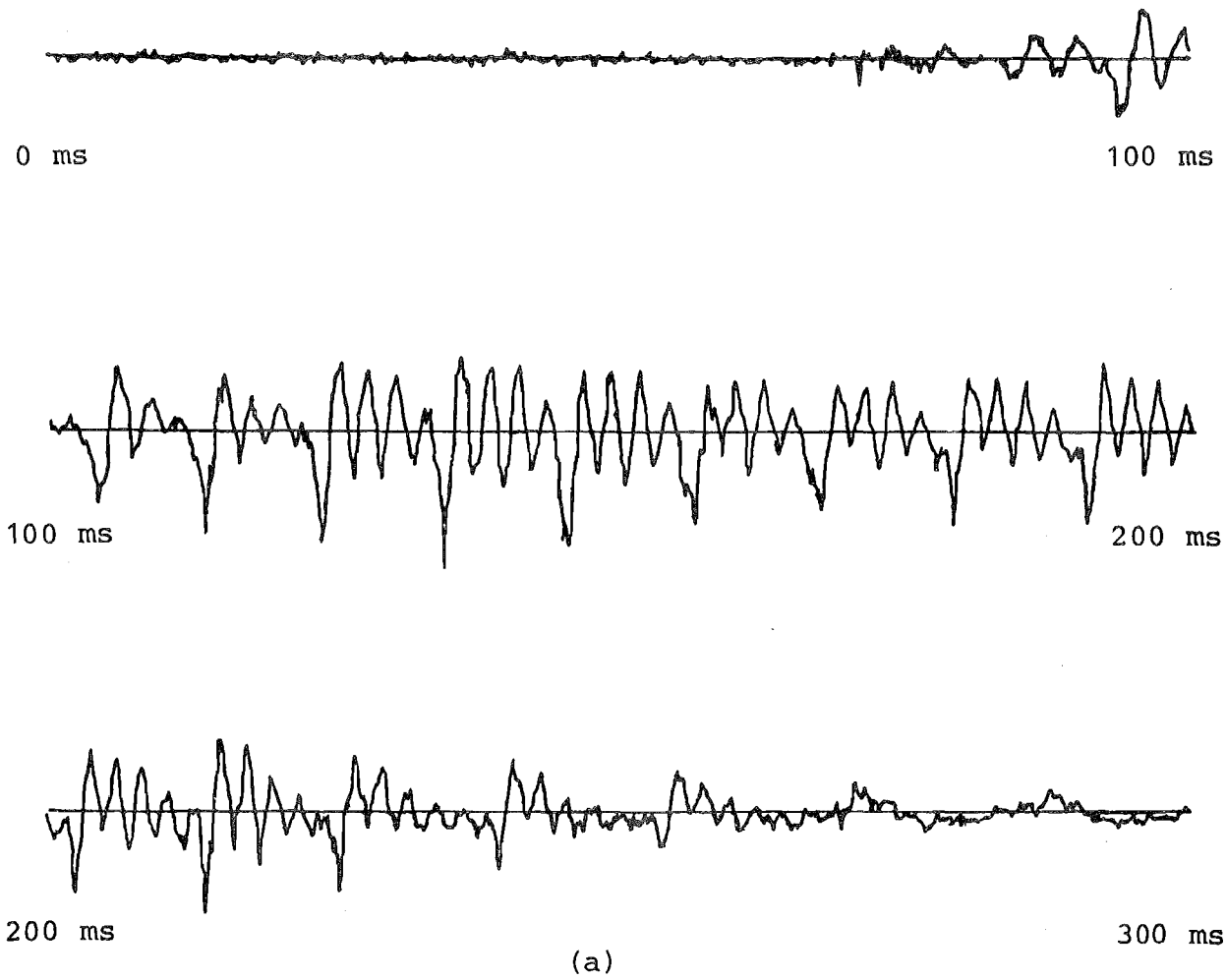
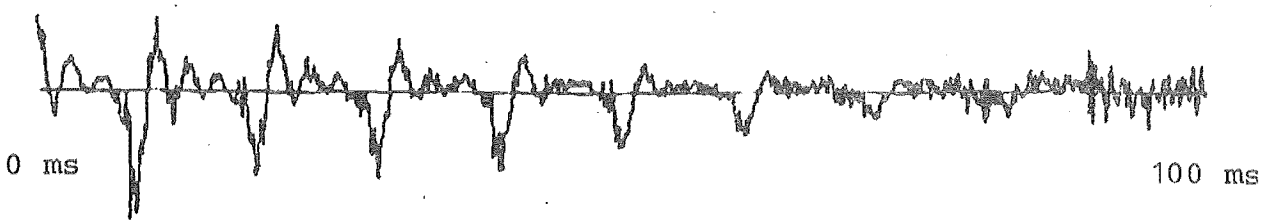
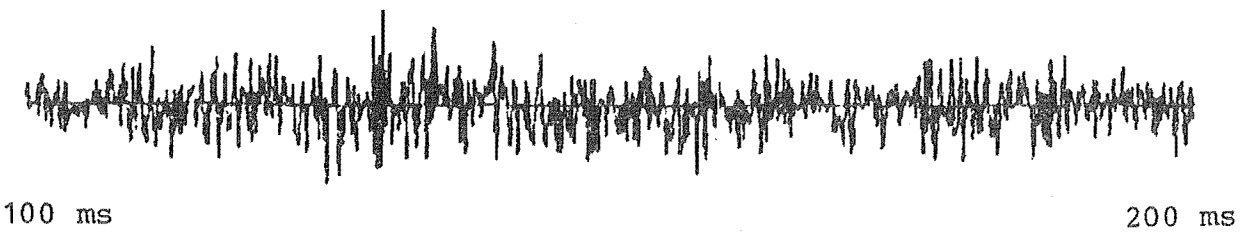


Figure 7.2 Speech and music waveforms:

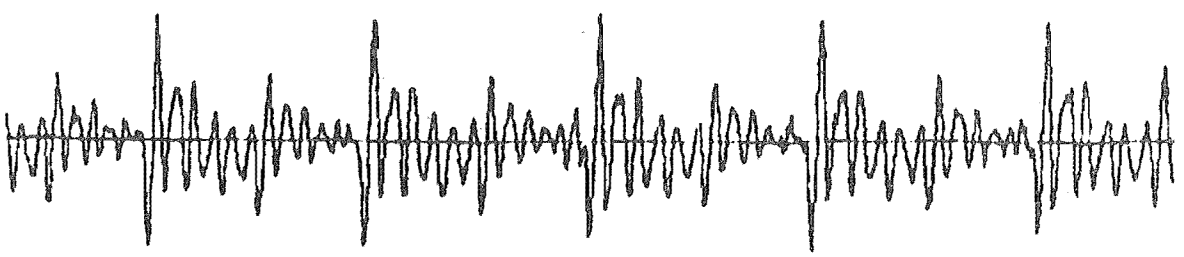
(a) A rapid vocal tract transition in speech (the voiced plosive /d/, male speaker).

(b) A "stationary" speech signal (the vowel /u/, male speaker).





(c)



(d)

Figure 7.2 Speech and music waveforms (continued):

(c) Voiced to unvoiced transition in speech. (Male speaker).

(d) Diplophonic speech, in which alternate periods are highly correlated, but adjacent periods are poorly correlated. (The vowel /*a*/, male speaker).

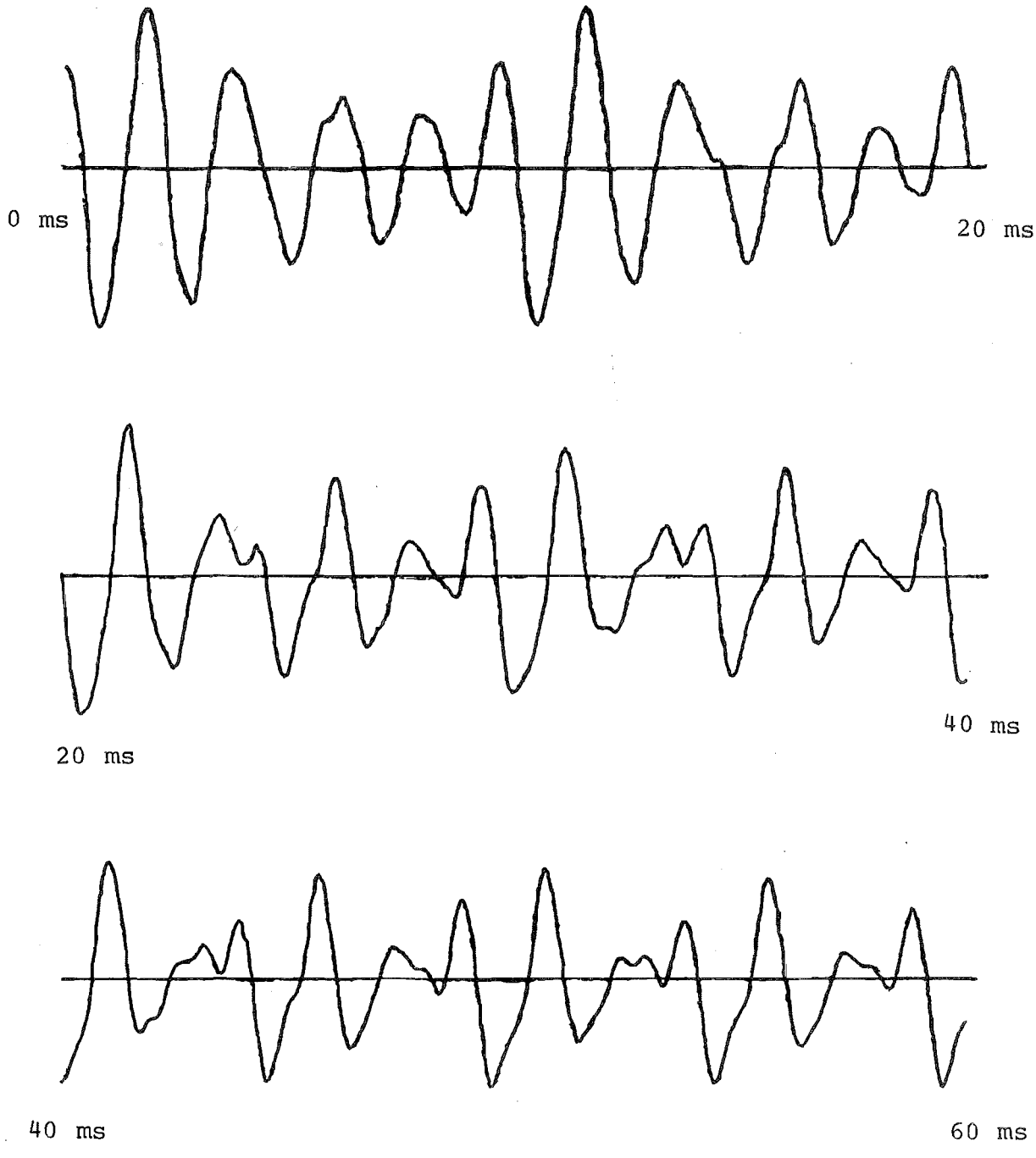
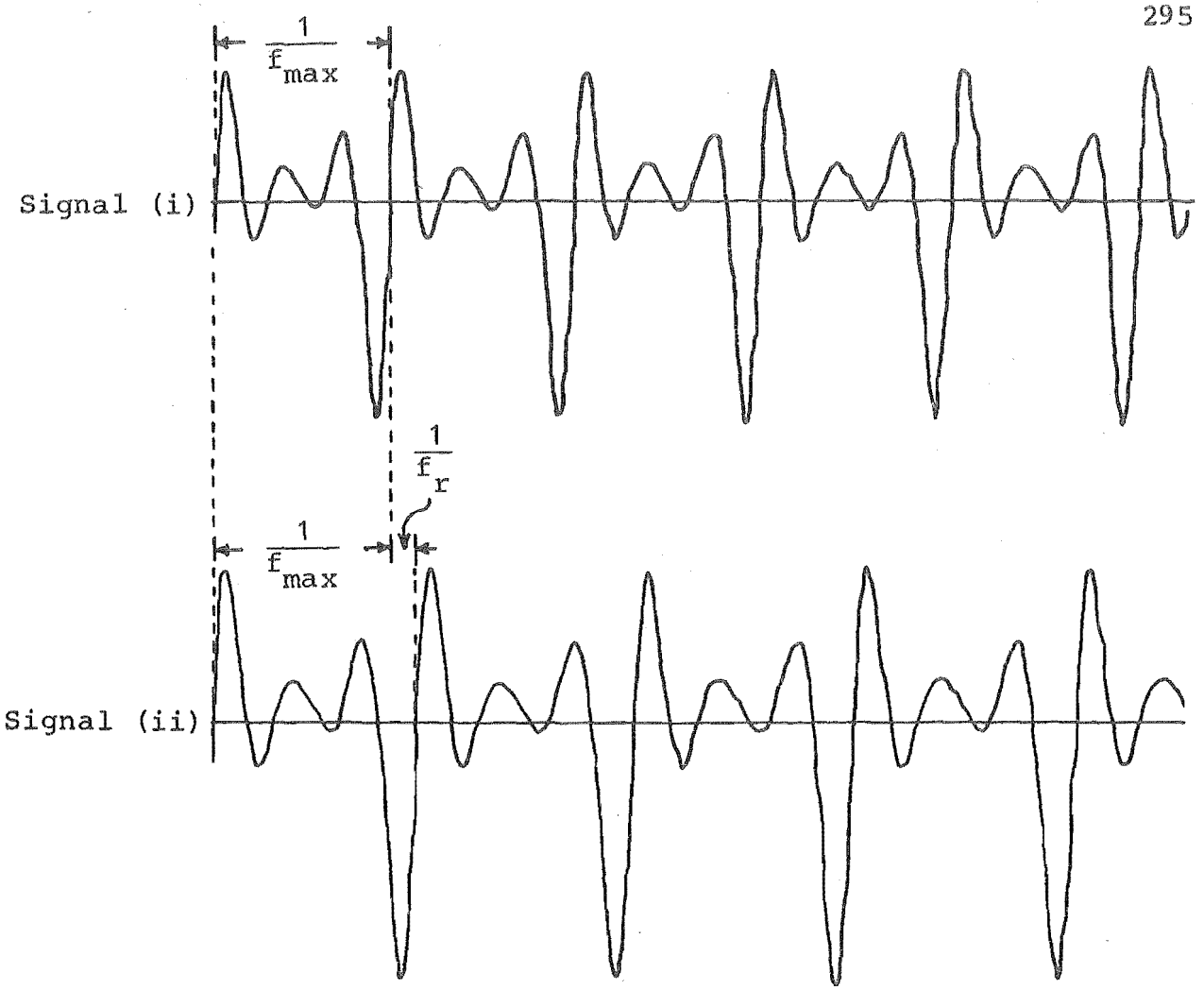
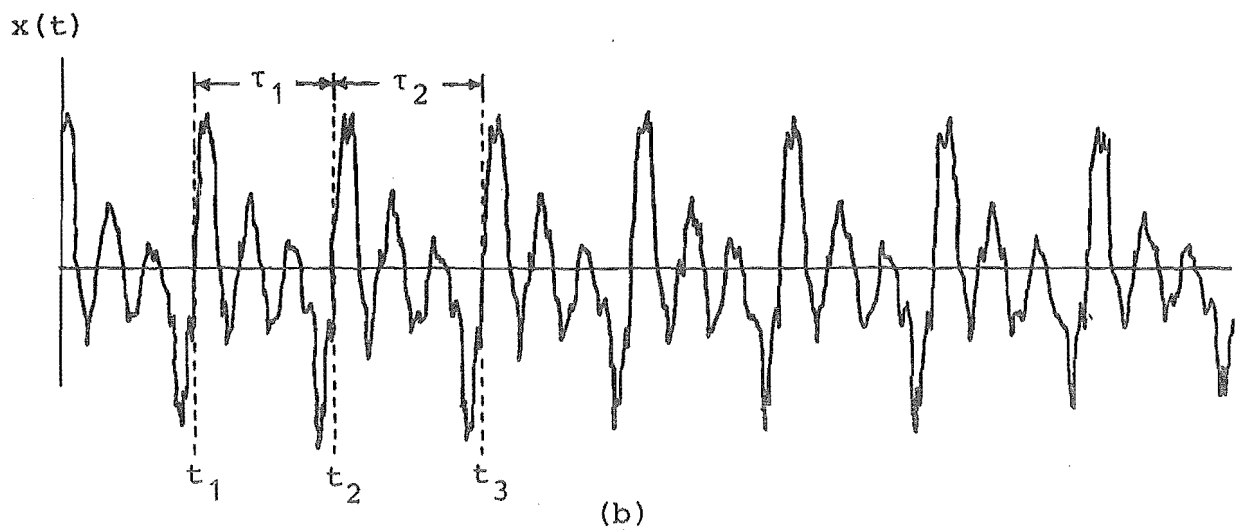


Figure 7.2 Speech and music waveforms (continued):

(e) Pitch doubling in the attack transient of a bassoon note.



(a)



(b)

Figure 7.3 (a) Illustrating the analysis of the interdependence between pitch frequency resolution, signal sampling rate and pitch frequency range (see Section 7.4.2).
(b) Illustrating the definitions used in Atal's (1968) analysis (see Section 7.4.4).

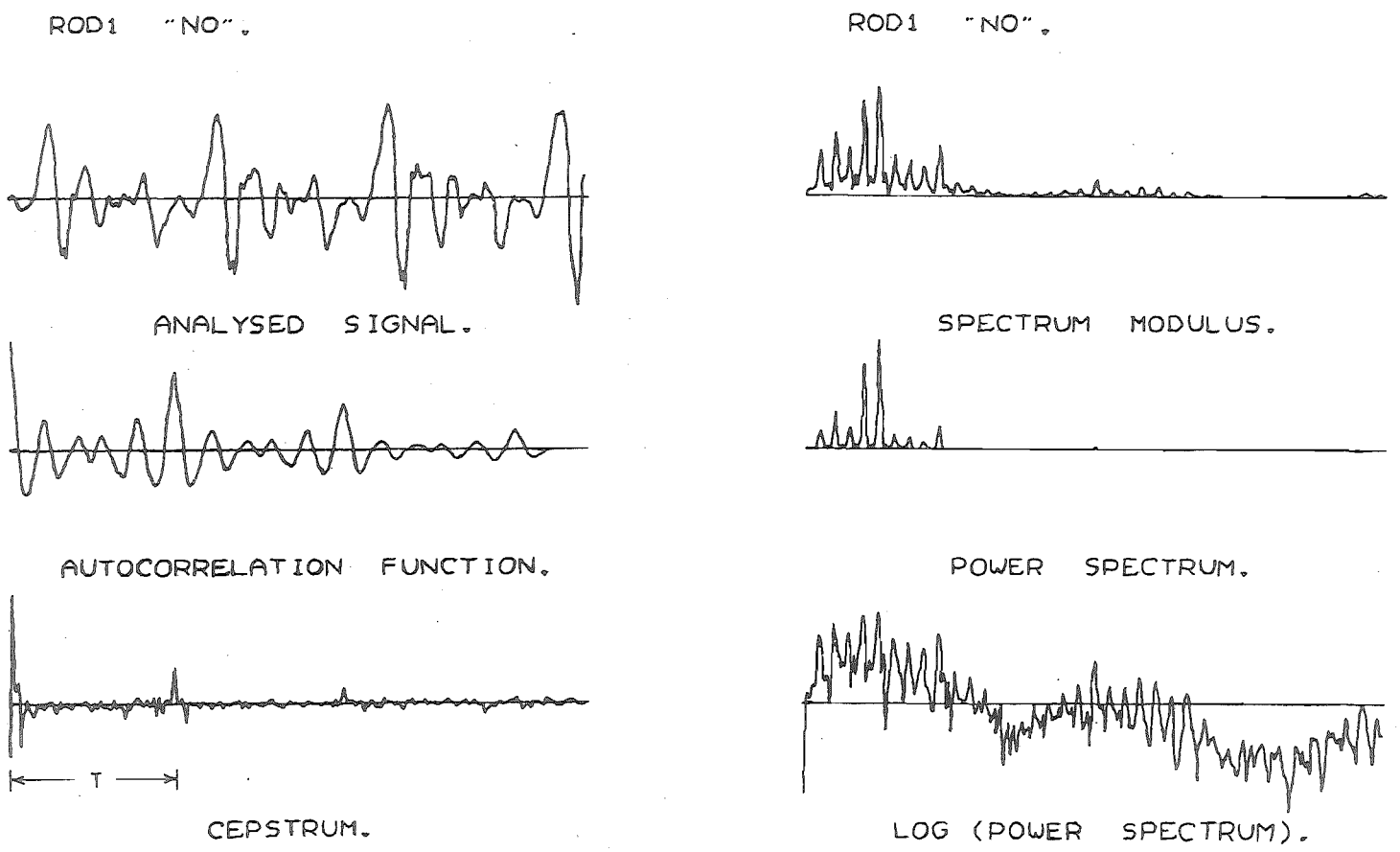


Figure 7.4 Comparing autocorrelation and cepstral analysis for speech, and illustrating the discussion of Section 7.5.3.

(a) Speech signal, its autocorrelation and its cepstrum.
(b) The spectrum, power spectrum and logarithm power spectrum of the speech signal.

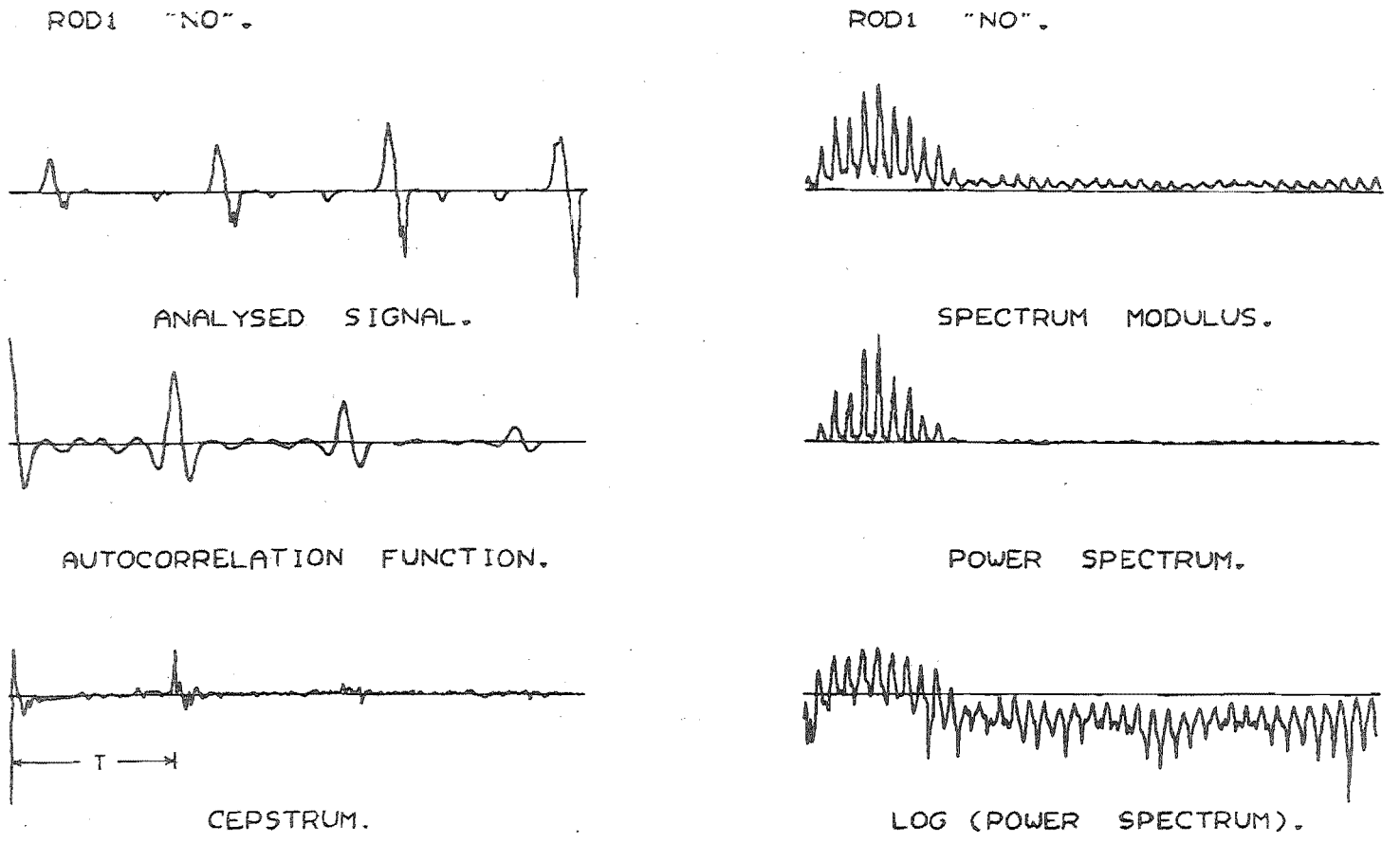


Figure 7.5 Illustrating the effect of signal cubing on the speech signal shown in Figure 7.4. Note that both the autocorrelation and cepstrum are improved.

(a) As for Figure 7.4(a).

(b) As for Figure 7.4(b).

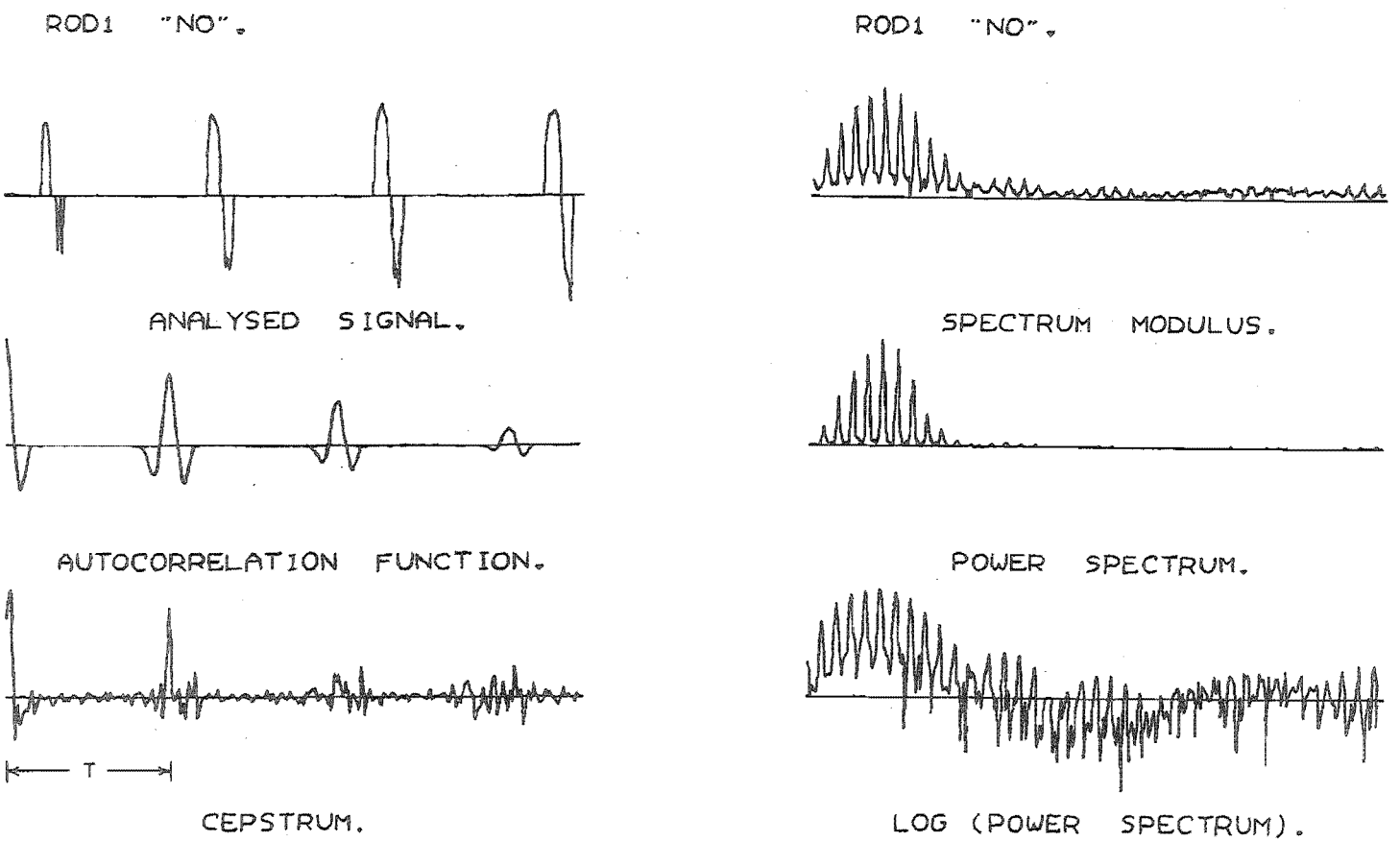


Figure 7.6 Illustrating the effect of centre clipping the signal shown in Figure 7.4. The clip level used is 50% of the absolute signal maximum. Observe that in this case the cepstral peak at quefrency T is very pronounced.

(a) As for Figure 7.4(a)

(b) As for Figure 7.4(b).

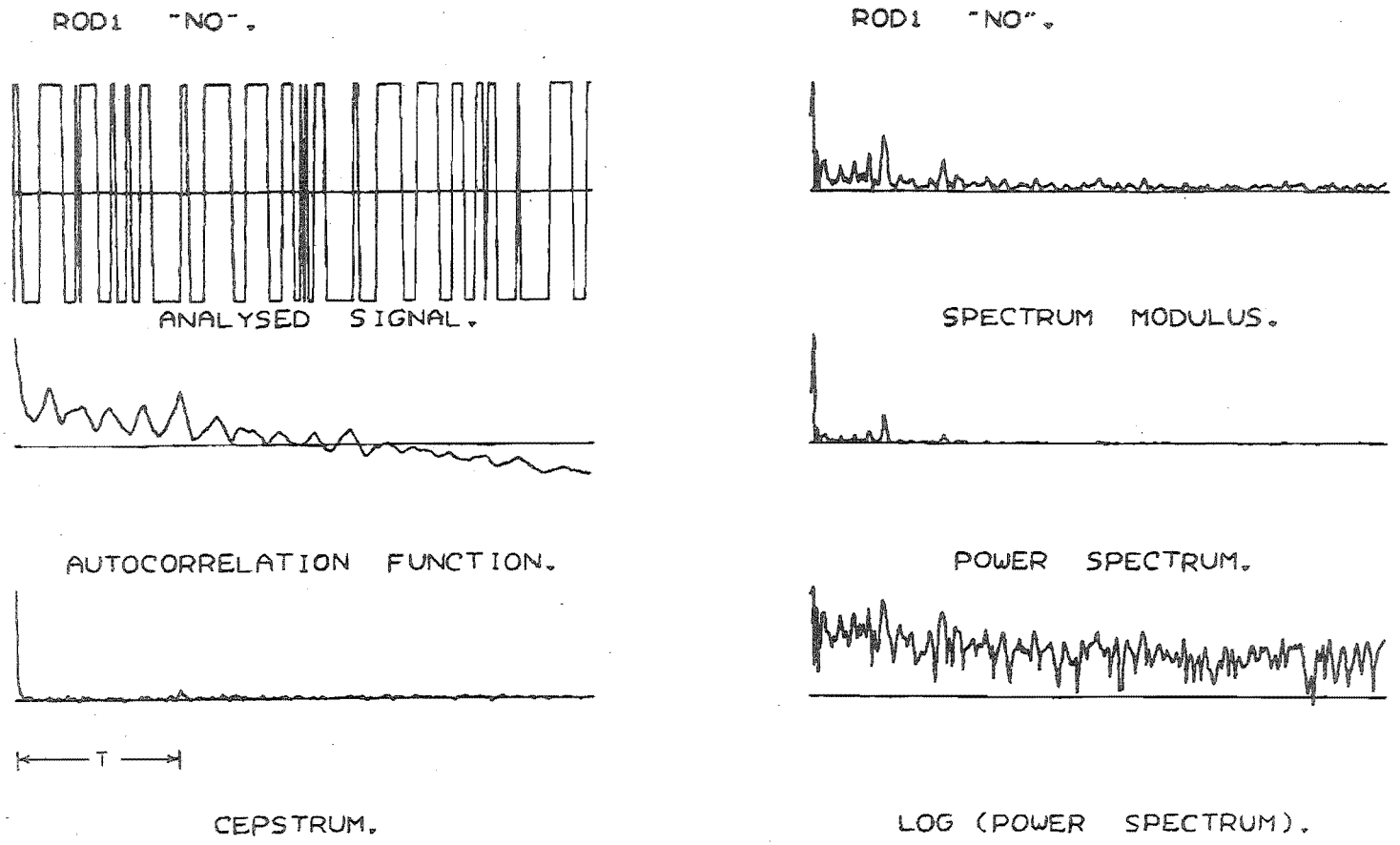


Figure 7.7 Illustrating the effect of infinite peak clipping on the signal shown in Figure 7.4. Observe that in this case both the autocorrelation and cepstrum provide poor indications of the signal period T.

(a) As for Figure 7.4(a)

(b) As for Figure 7.4(b).

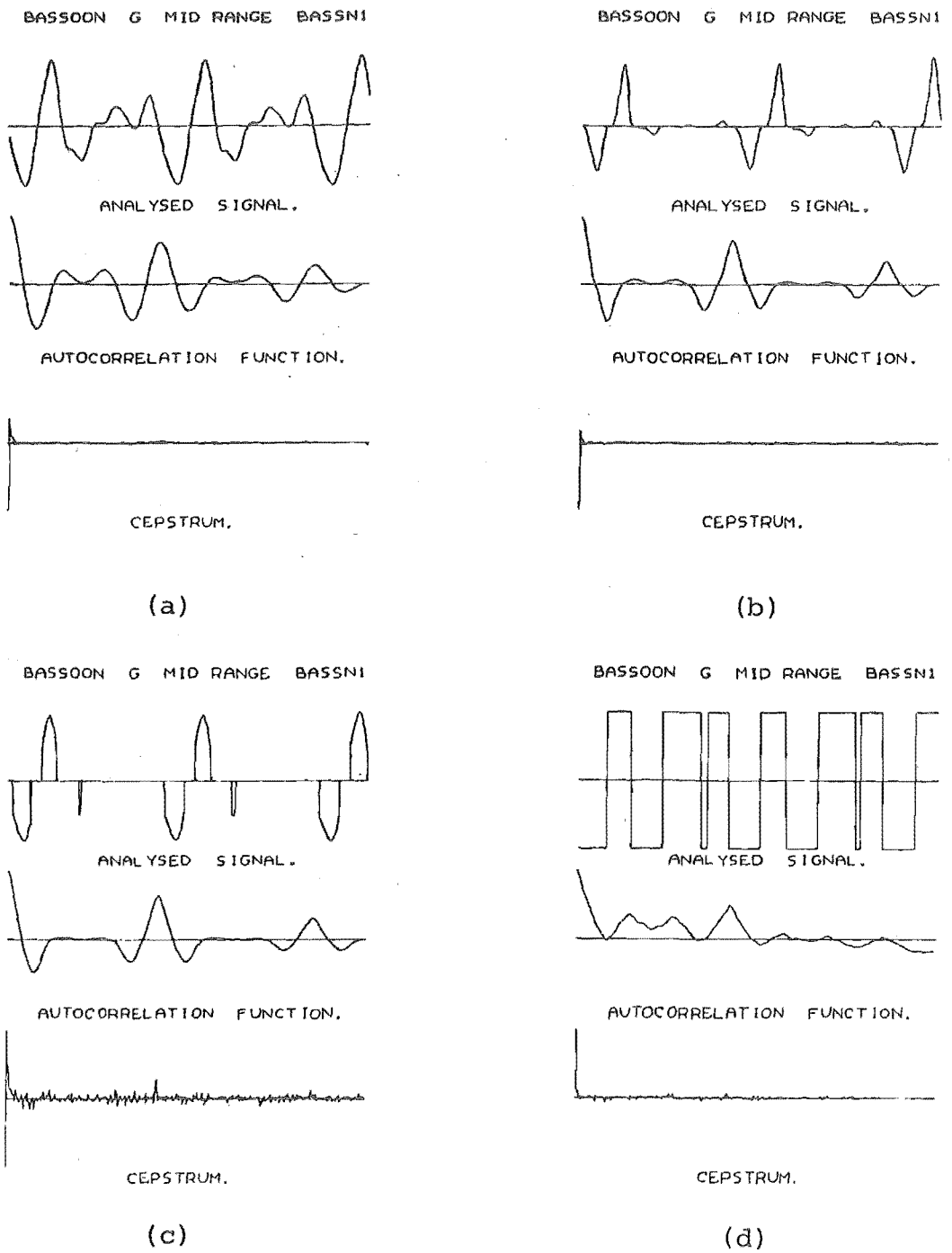


Figure 7.8 Comparison of autocorrelation and cepstrum for typical BASSOON waveform.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping

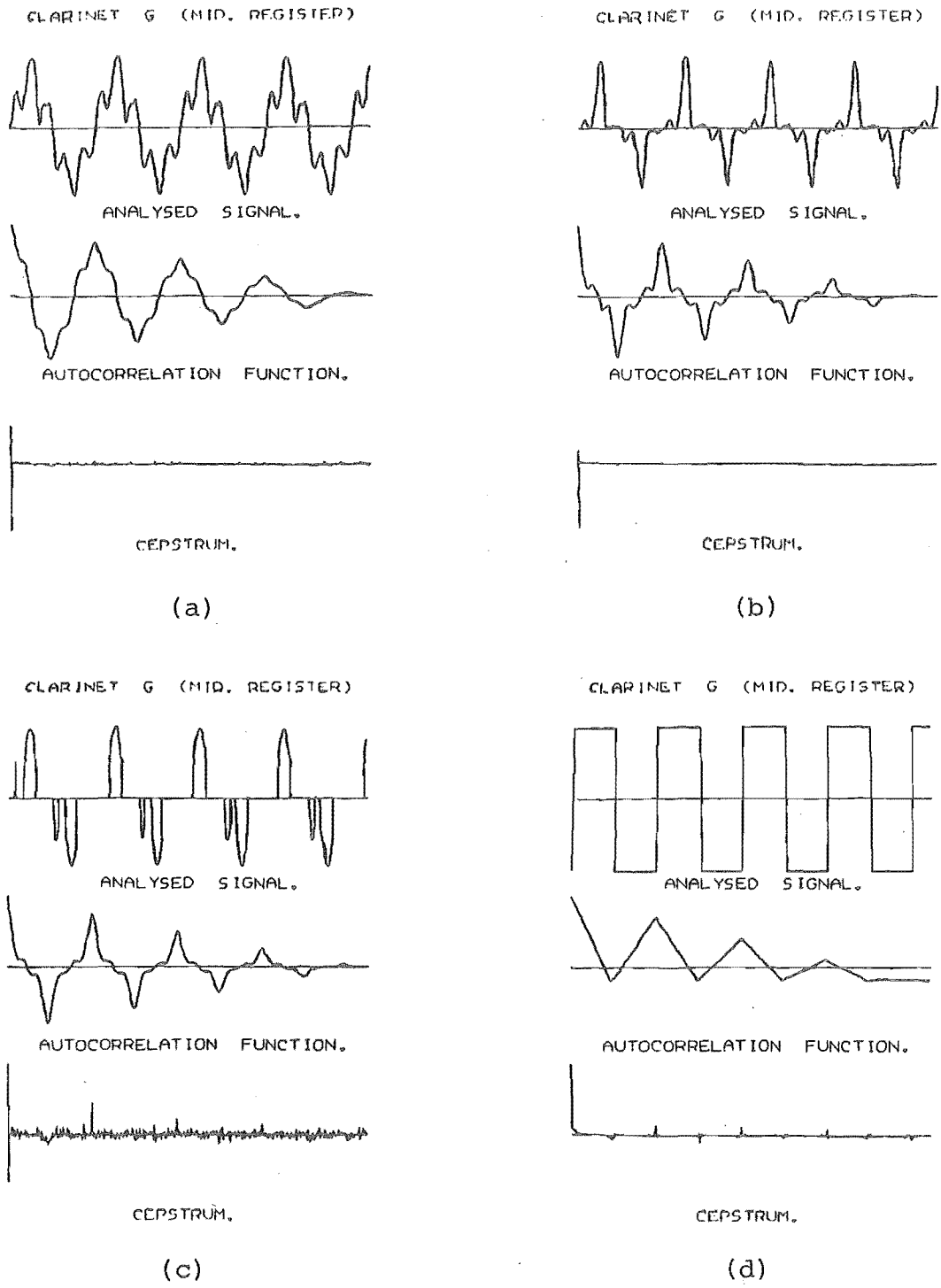


Figure 7.9 Comparison of autocorrelation and cepstrum for typical CLARINET waveform.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

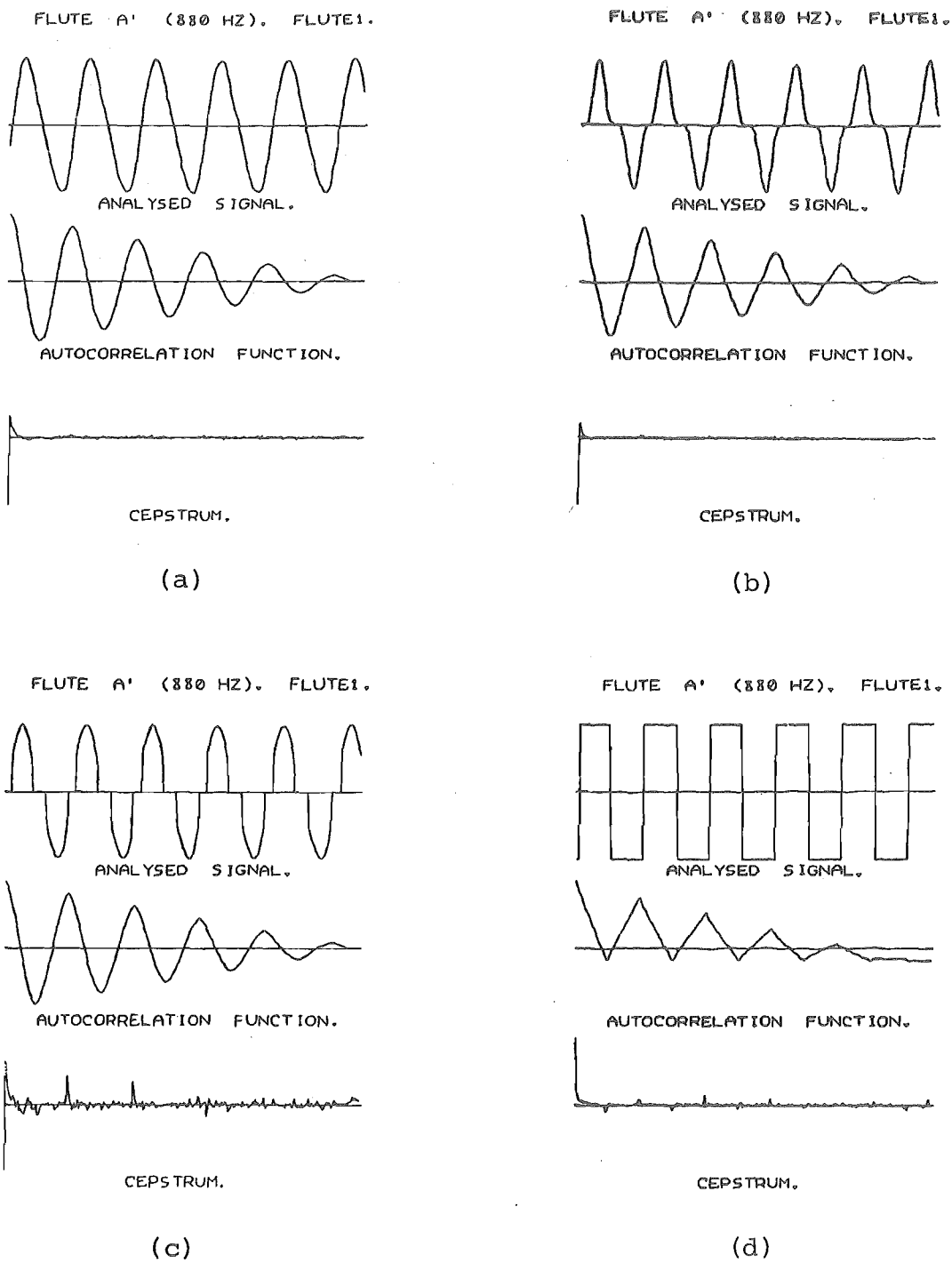


Figure 7.10 Comparison of autocorrelation and cepstrum for typical FLUTE waveform.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

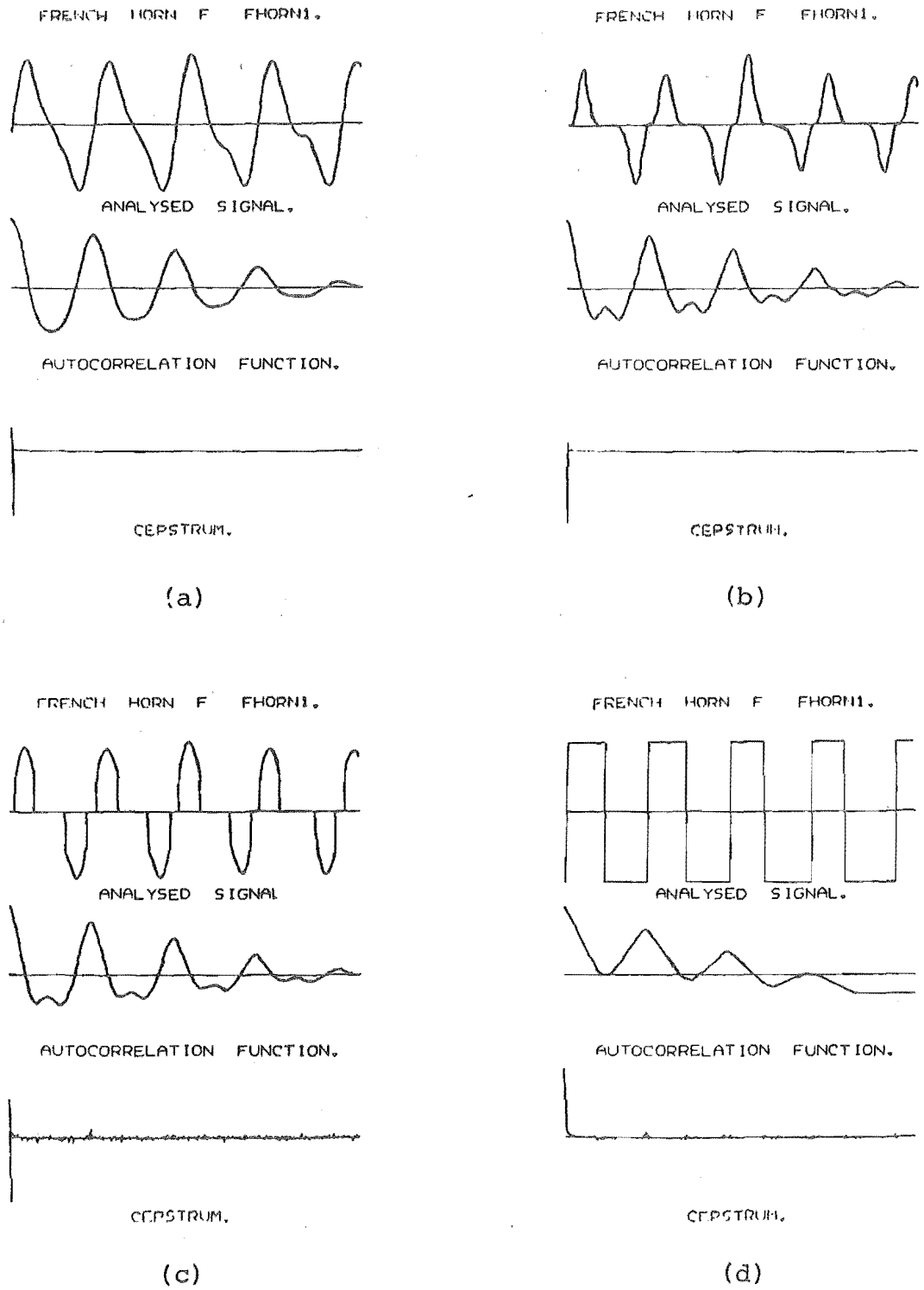
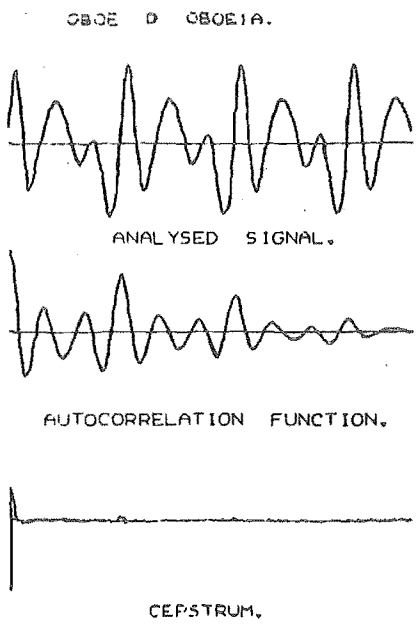
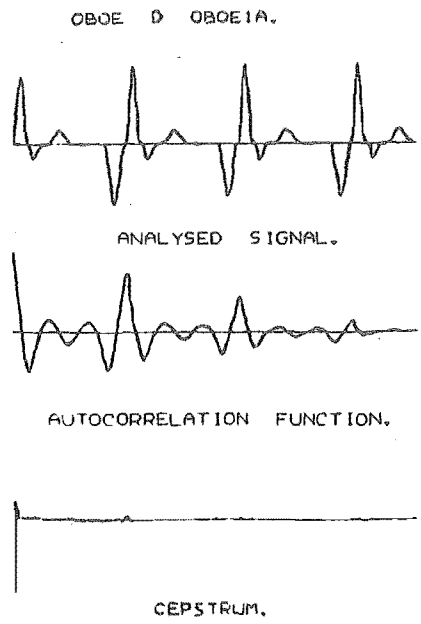


Figure 7.11 Comparison of autocorrelation and cepstrum for typical FRENCH HORN waveform.

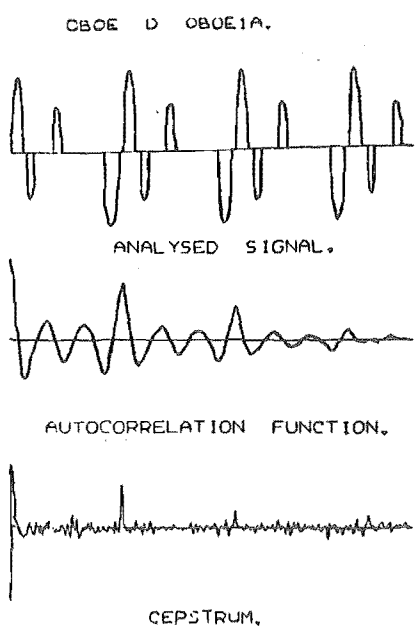
(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.



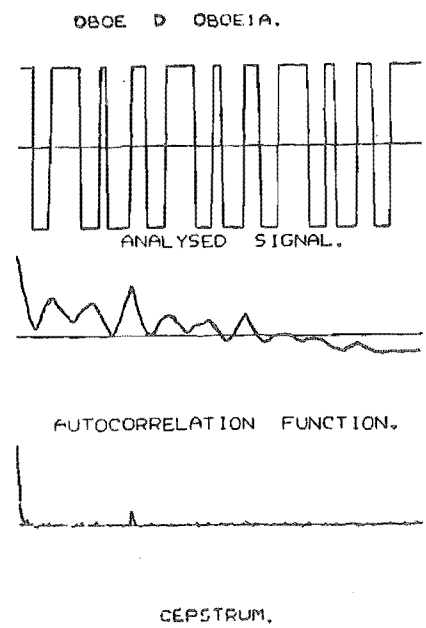
(a)



(b)



(c)



(d)

Figure 7.12 Comparison of autocorrelation and cepstrum for typical OBOE waveform.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

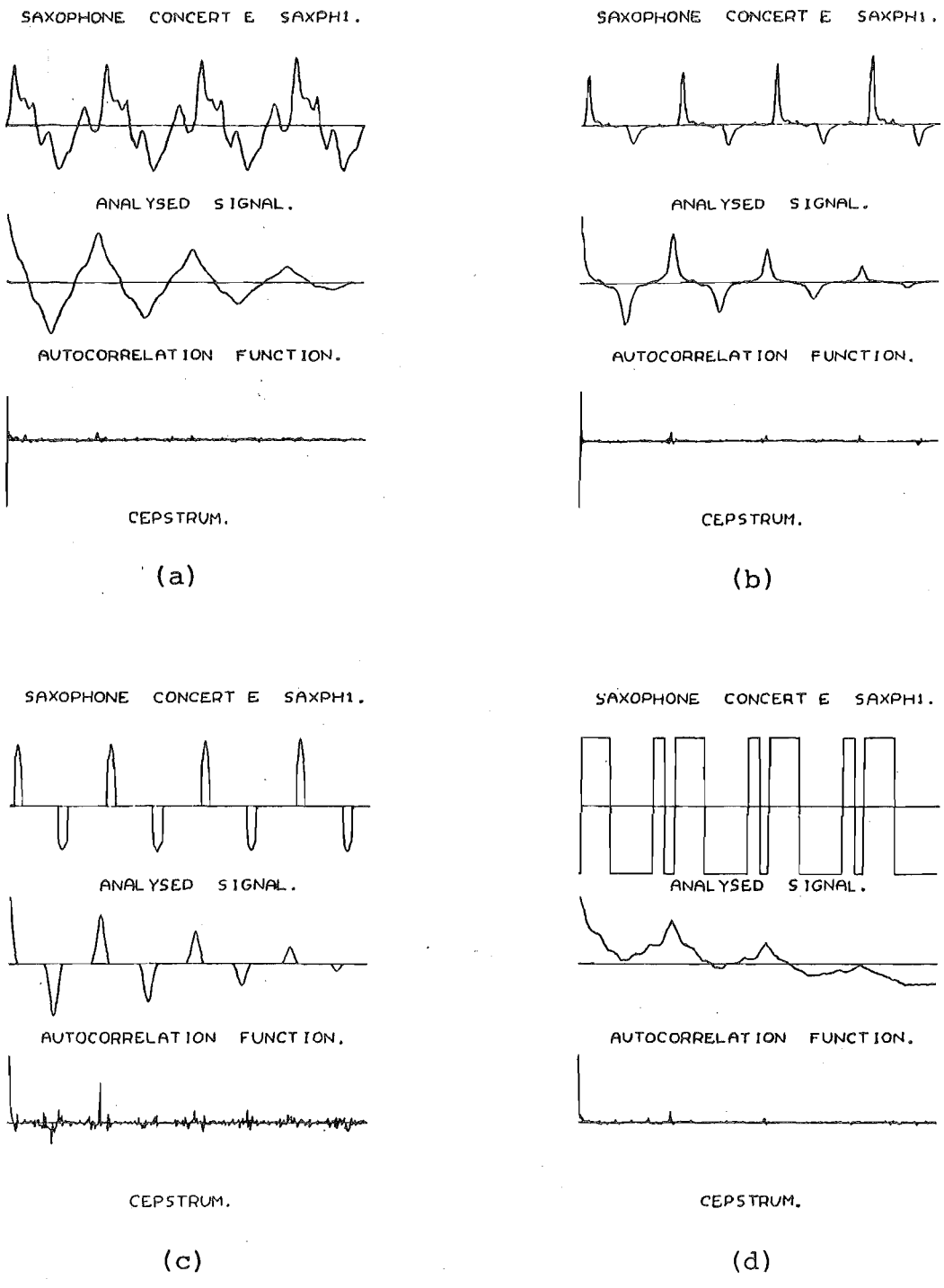


Figure 7.13 Comparison of autocorrelation and cepstrum for typical SAXOPHONE waveform.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

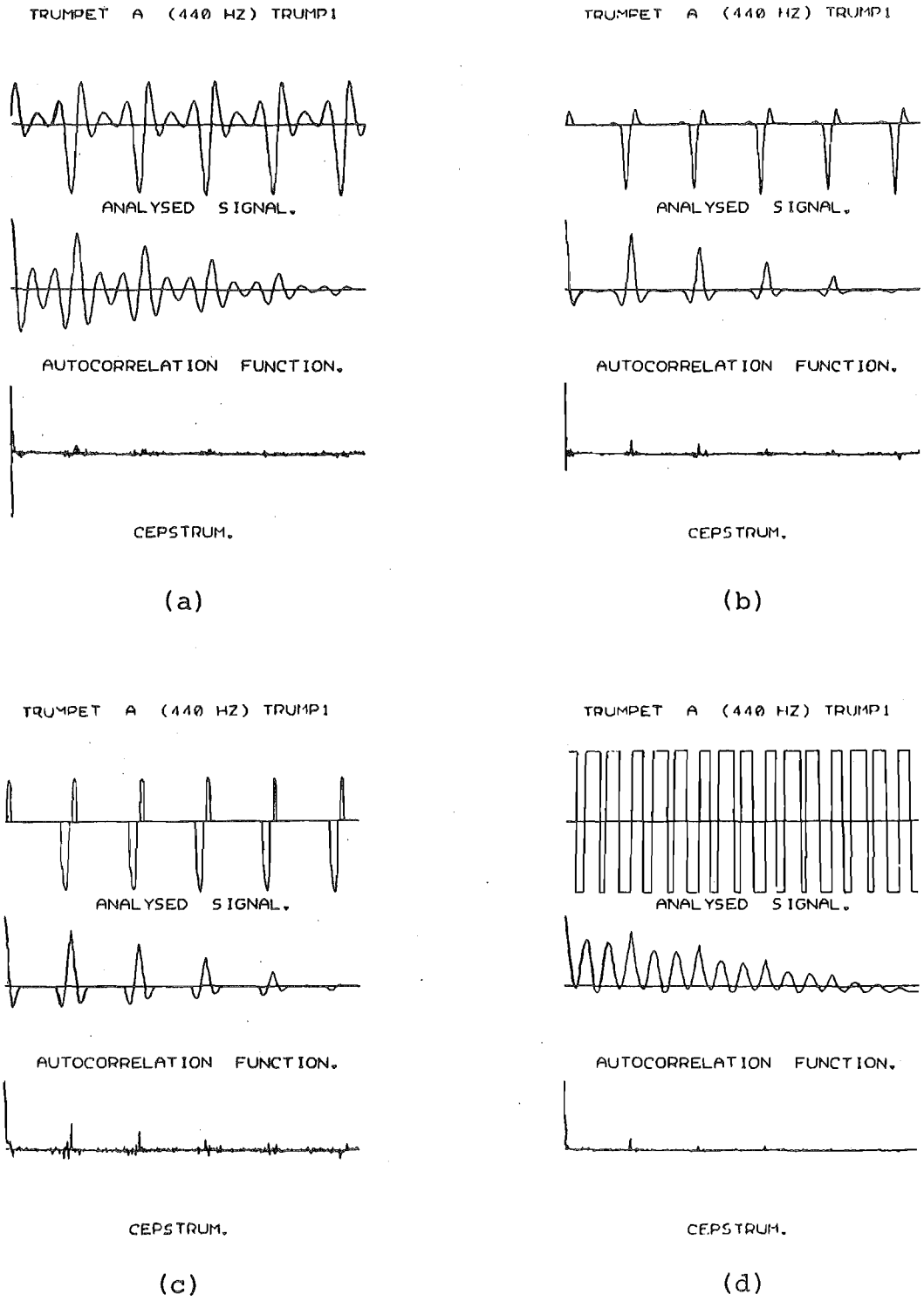


Figure 7.14 Comparison of autocorrelation and cepstrum for typical TRUMPET waveform.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

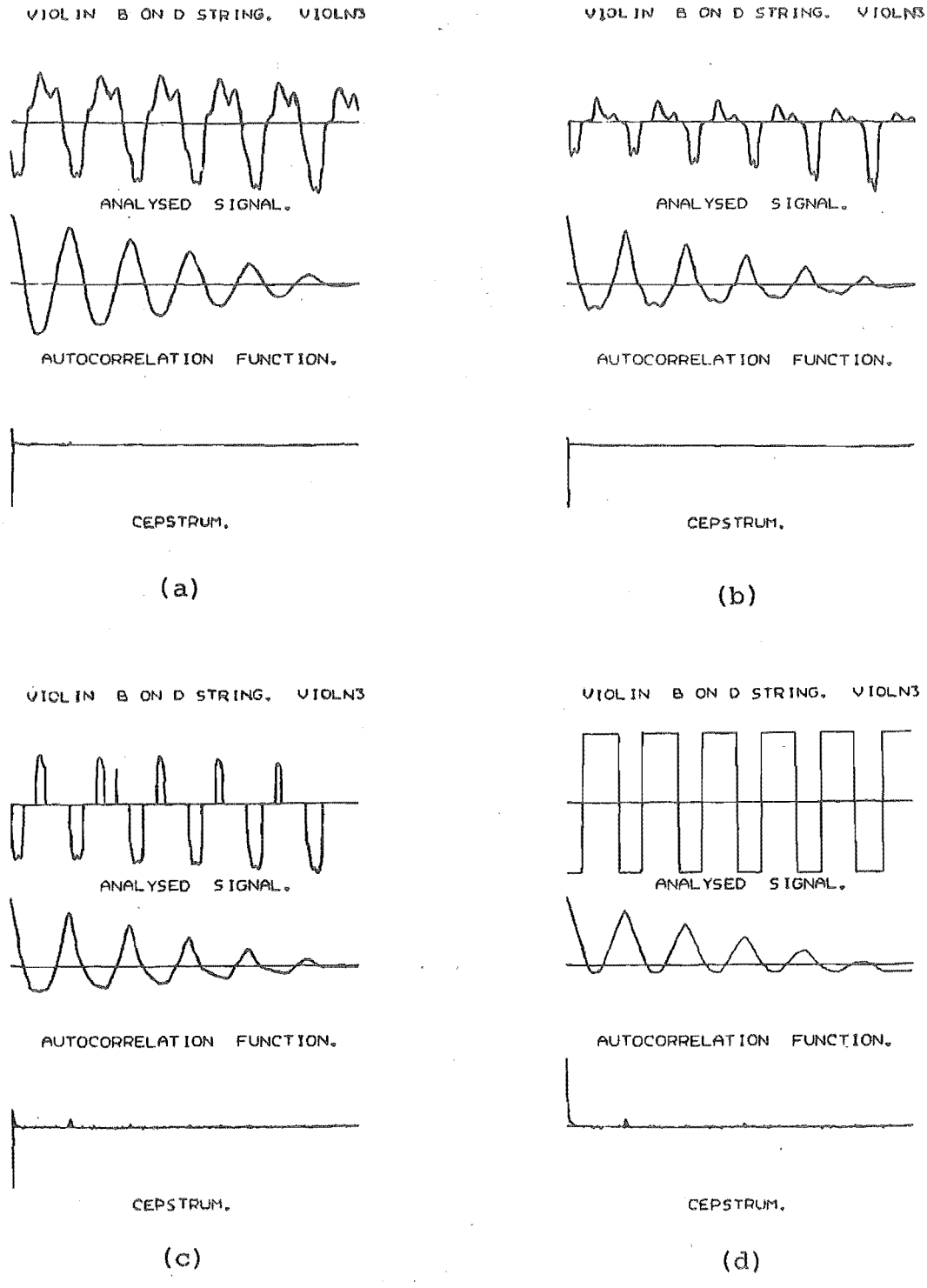


Figure 7.15 Comparison of autocorrelation and cepstrum for typical VIOLIN waveform.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

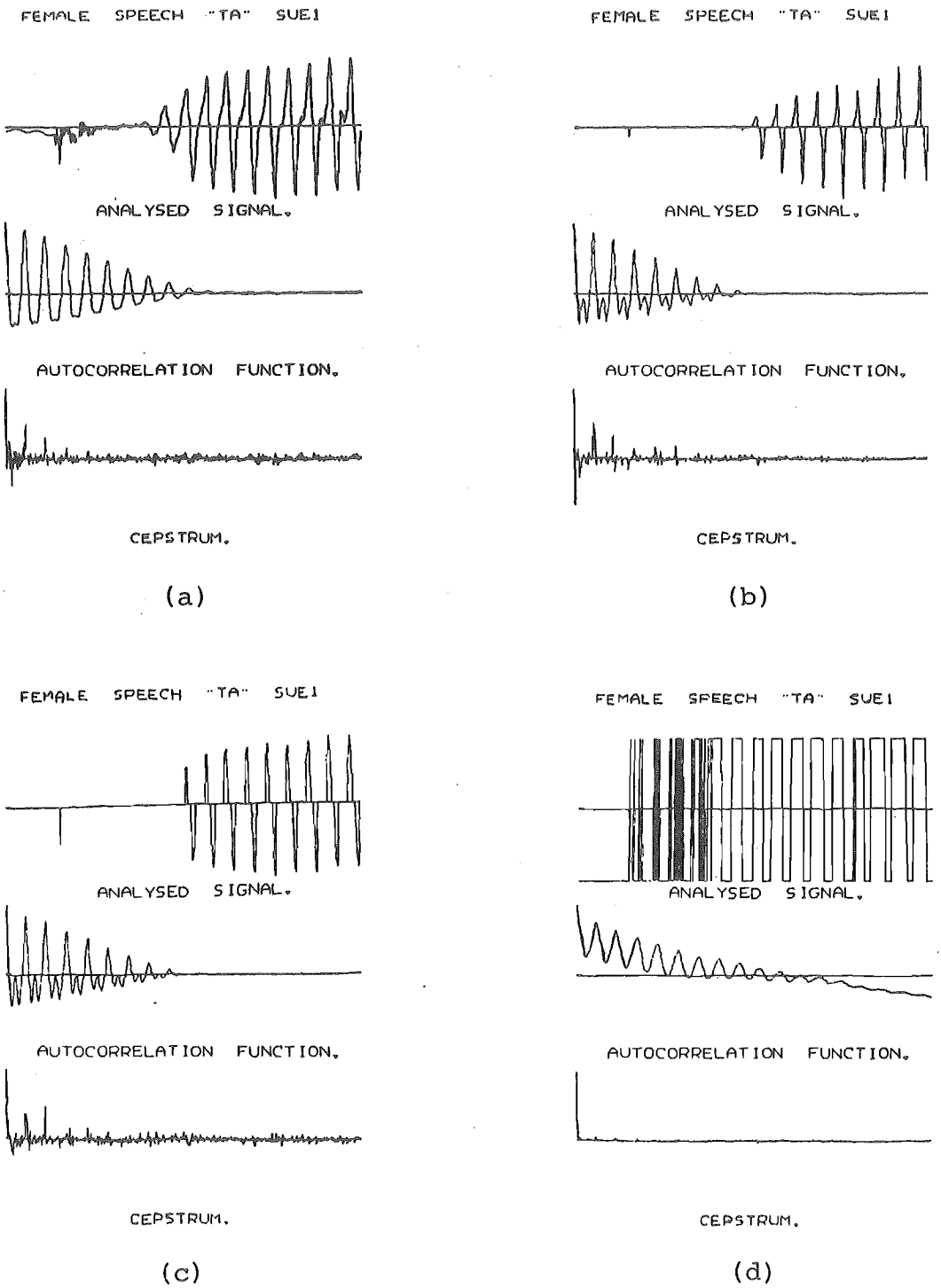


Figure 7.16 Comparison of autocorrelation and cepstrum for a typical high-pitched speech sound ("ta") with a long analysis window.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

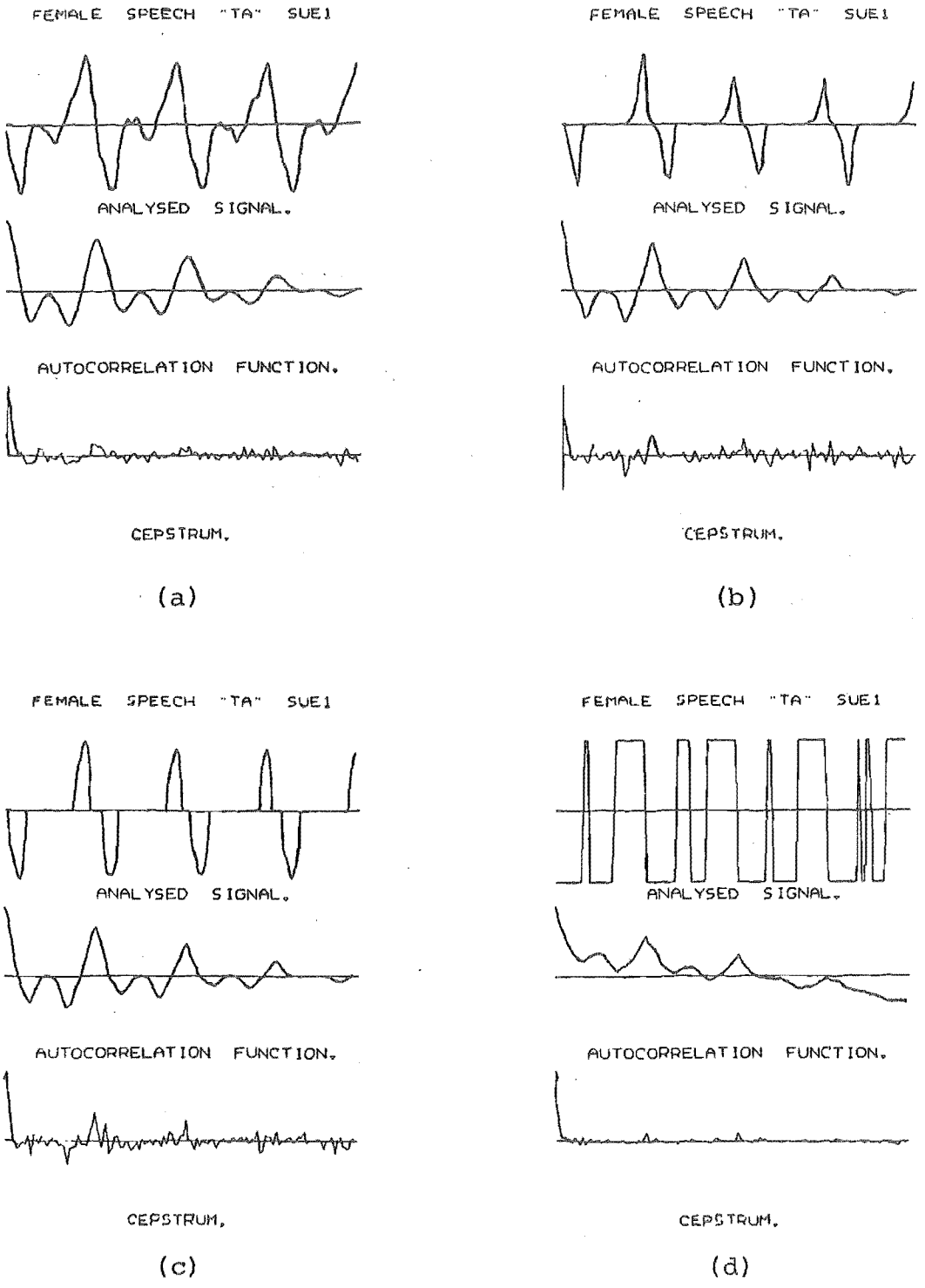


Figure 7.17 Comparison of autocorrelation and cepstrum for a short segment of the signal shown in Figure 7.16.

(a) Original signal
(b) Preprocessed by cubing
(c) Preprocessed by centre clipping
(d) Preprocessed by infinite peak clipping.

(a)

(b)

(c)

Figure 7.18 Typical Laryngograph waveforms (after Fourcin and Abberton, 1971).

(a) Normal voiced speech
(b) Voiced speech of subject with unilateral palsy
(c) Voiced speech of subject with laryngitis.

CHAPTER 8

TRANSFORM METHODS FOR CORRELATION - A REVIEW

8.1 TRANSFORMS, CONVOLUTION AND CORRELATION

The use of the Fourier transform to compute the autocorrelation function of a signal has already been mentioned (see Section 7.4.2). This frequency domain approach is possible because of the convolution theorem possessed by the Fourier transform. The relationship between convolution and correlation is considered in the standard texts (e.g. Lathi, 1965) and need not be repeated here. It is sufficient to state the Autocorrelation theorem (which is also called the Wiener-Khinchin theorem):

$$\phi(\tau) = s(t) * s(-t) \leftrightarrow S(\omega)\, S(-\omega) = |S(\omega)|^2 \qquad (8.1)$$

where $\phi(\tau)$ is the autocorrelation function of the real signal $s(t)$, the operator * denotes convolution and $s(t)$ and $S(\omega)$ are Fourier transform pairs.

This chapter briefly reviews the use of transform methods to numerically evaluate convolutions and correlations. While the emphasis here is on autocorrelation, most of the discussion applies more generally to correlation and convolution. The only major exception to this occurs in Section 8.4.3, where the computation of the autocorrelation function using Walsh functions is considered - this

discussion is specific to autocorrelation. Note also that the terms "convolution" and "correlation" are herein used interchangeably, since the only difference between them from a computation viewpoint is in the ordering of the input data sequences (see for example Section 12.2 of Lathi, 1965). Provided this is clearly understood it need cause no confusion.

Section 8.2 outlines the use of the FFT to compute non-cyclic convolution. In Section 8.3 the number theoretic transform is considered, and the Rader transform (RT), Mersenne Number transform (MNT) and Fermat number transform (FNT) are discussed. Methods based on the Walsh transform are presented in Section 8.4. These include the method of Pitassi (1971) and its generalisation by Davis (1972) as well as the transformation from the dyadic to arithmetic autocorrelation.

8.2 FAST FOURIER TRANSFORM

The numerical evaluation of $\phi(\tau)$ using the approach suggested by the Autocorrelation theorem (equation 8.1) is conveniently performed using the FFT, as Stockham (1966) first suggested. Computational considerations are discussed in Section 7.4.2, where it is pointed out that this method is more efficient than direct time domain computation if $\phi(\tau)$ is required for many values of τ, or if the sequence of signal samples $\{s_n\}$ is long. This computational superiority is due to the fact that direct time domain evaluation of $\phi(\tau)$ (as defined by equation 7.7) requires approximately $(N/4)(N/2 + 3)$ multiplications, whereas the

FFT method requires approximately $3N \log_2 N$ multiplications if a radix 2 FFT is used (Cochran *et al.*, 1967). N here is the length of the sequence $\{s_n\}$. These figures are only approximate and vary with different implementations (cf. Section 7.4.2): however, they provide a useful indication of the relative computation requirements. Measurements by Gethöffer (1973) of the total computation time for both the direct and FFT methods indicate that for values of N greater than $2^8 = 256$ the FFT method is faster than direct computation of $\phi(\tau)$. This conclusion takes into account the frame length doubling which is required when non-cyclic convolution is performed by the (cyclic) FFT. The use of an FFT algorithm which employs pruning (Markel, 1971; Skinner, 1976) to eliminate unnecessary operations on the zero data terms makes the FFT method even more attractive.

Many papers which exist in the literature discuss the use of the FFT to compute correlations. A useful collection of these is given by Rabiner and Rader (1972). The texts by Oppenheim and Schafer (1975) and Rabiner and Gold (1975) provide additional useful information.

Recently Rader and Brenner (1976) have derived secant and cosecant forms of the FFT, which have the property that none of the multiplying factors are complex (most are pure imaginary). These forms are useful if complex sequences are being convolved using hardware for which multiplication is slow and/or costly. However, they are more susceptible to accumulated computation errors than are the conventional FFT forms.

Winograd (1976) derives a new algorithm - the WFTA -

which permits the DFT to be evaluated using fewer multiplications and approximately the same number of additions as required by the FFT. Silverman (1977) describes various forms of the WFTA, and discusses them from a programming and computational viewpoint. Silverman also presents a comparison between the WFTA and the FFT, and shows that the former exhibits a speed advantage of the order of 1.5 to 2.5 over the latter. For a detailed description of the WFTA the reader is referred to Silverman (1977).

A useful software technique for generating fast in-line FFT machine code is described by Morris (1977). This technique eliminates the time-consuming decision making and arithmetic operations for loop control or data access, which for conventional software implementations are performed during execution. Morris observes that this approach can result in transformation speed improvement by a factor of 2 to 3.

8.3 NUMBER THEORETIC TRANSFORMS

The FFT operates on sequences in the complex number field. However, in practical applications the data sequences are available with only finite precision. Consequently, without loss of generality, the data can be considered to be integers with some upper bound. In this digital domain, the complex number field of the continuous domain can be replaced by a finite field, or more generally by a finite ring of integers for which multiplication and addition are defined modulo some integer F. In this ring

cyclic convolution can be performed very efficiently using transforms similar to the FFT. These transforms are called "number theoretic transforms", and permit the cyclic convolution to be computed without the roundoff error which is obtained when the FFT is used (Agarwal and Burrus, 1974).

The use of transforms in finite fields is proposed by Knuth (1969). Good (1971) points out the existence of number theoretic transforms which possess a convolution theorem. Pollard (1971) discusses transforms which have the cyclic convolution property in a finite field, and gives the conditions required by such transforms in a finite ring of integers. Agarwal and Burrus (1974) cite the work by Schonhage and Strassen (1971), Knuth (1971) and Nicholson (1971). The application of number theoretic transforms to digital signal processing is proposed by Rader (1972a, 1972b), whose main motivation was the desire to eliminate the truncation and roundoff errors encountered during convolution by transforms in the complex number field. He proposes finite transforms in rings of integers modulo both Mersenne and Fermat numbers and shows that such transforms can be calculated using only addition and bit shifting (cf. the FFT which requires both addition and multiplication). He also shows that a constraint is imposed on the processor word length and suggests the use of a two-dimensional transform to overcome this constraint when long sequences are to be convolved. These points are discussed in Section 8.3.2.

Agarwal and Burrus (1974) examine the structure of transforms which have the cyclic convolution property.

They consider the class of linear invertible (i.e. non-singular) transforms which map a sequence of length N to another sequence of length N. The main conclusion from this examination is that the "DFT structure" is the only structure which supports the cyclic convolution property, and that any transform which possesses this structure will have the cyclic convolution property. Since the approach used by Agarwal and Burrus provides a useful insight to the nature of such transforms it is summarised below.

Let $\{x_n\}$ and $\{h_n\}$, $n = 0, 1, 2, \ldots N-1$, be two sequences which are to be cyclically convolved. Denote by $\{y_n\} = \{x_n\} * \{h_n\}$ the convolution of x_n and h_n. Then

$$y_n = \sum_{k=0}^{N-1} x_k \, h_{n-k} \qquad n = 0, 1, \ldots N-1 \tag{8.2}$$

where the sequences are periodically extended (with period N) or, equivalently, the indices are evaluated modulo N. It is convenient to invoke the latter requirement, and to consider the sequences as vectors of length N. Since only linear invertible transforms are considered, they can be represented by an $N \times N$ nonsingular matrix T whose elements are $t_{k,m}$, $k,m = 0, 1, \ldots N-1$. Denote transformed sequences by capital letters, so that

$$\begin{aligned} \underset{\sim}{X} &= T\underset{\sim}{x} \\ \underset{\sim}{H} &= T\underset{\sim}{h} \\ \underset{\sim}{Y} &= T\underset{\sim}{y} \; . \end{aligned} \tag{8.3}$$

Consider the conditions on the $t_{k,m}$'s so that

$$\underset{\sim}{Y} = \underset{\sim}{X} \otimes \underset{\sim}{H} \tag{8.4}$$

where $\otimes$ denotes term by term multiplication of the vectors. Combining equations (8.2) and (8.3) gives

$$Y_k = \sum_{n=0}^{N-1} t_{k,n}\, y_n = \sum_{n=0}^{N-1} t_{k,n} \sum_{m=0}^{N-1} x_m\, h_{n-m}$$

$$= \sum_{m=0}^{N-1} \sum_{\ell=0}^{N-1} x_m\, h_\ell\, t_{k,m+\ell}$$

$$X_k = \sum_{m=0}^{N-1} t_{k,m}\, x_m$$

$$H_k = \sum_{\ell=0}^{N-1} t_{k,\ell}\, h_\ell \qquad k = 0,\ 1,\ \ldots N-1\ . \tag{8.5}$$

The individual terms in (8.4) can be rewritten as

$$Y_k = X_k\, H_k$$

which, using (8.5), becomes

$$\sum_{m=0}^{N-1} \sum_{\ell=0}^{N-1} x_m h_\ell t_{k,m+\ell} = \sum_{m=0}^{N-1} \sum_{\ell=0}^{N-1} x_m h_\ell t_{k,m} t_{k,\ell}\ . \tag{8.6}$$

Matching the multiples of $x_m h_\ell$ on both sides of equation (8.6) gives

$$t_{k,m+\ell} = t_{k,m}\, t_{k,\ell} \qquad k,_\ell,m = 0,\ 1,\ \ldots N-1 \tag{8.7}$$

Repeated applications of equation (8.7) gives

$$t_{k,m} = t_{k,1}^{\,m} \qquad k,m = 0,\ 1,\ \ldots N-1 \tag{8.8}$$

and since the convolution is cyclic, the indices in equation

(8.7) are added modulo N. This gives

$$t_{k,m}^{N} = 1 \qquad k,m = 0, 1, \ldots N-1 \qquad (8.9)$$

Therefore the $t_{k,m}$'s are the N^{th} roots of unity. For T to be nonsingular, all the $t_{k,1}$'s must be distinct. Since there are only N distinct roots of unity, the $t_{k,1}$'s must be these N distinct roots. Without loss of generality, $t_{1,1}$ can be taken as a root of order N, so that N is the least positive integer which satisfies $t_{1,1}^{N} = 1$. Therefore all the $t_{k,1}$'s can be written as some power of $t_{1,1}$. Again without loss of generality, the rows of T can be arranged so that

$$t_{k,1} = t_{1,1}^{k} . \qquad (8.10)$$

Combining equations (8.8) and (8.10), and denoting $t_{1,1}$ by α gives

$$t_{k,m} = \alpha^{km} \qquad k,m = 0, 1, \ldots N-1 . \qquad (8.11)$$

The structure established in this way causes the transform to be orthogonal. The elements $\tilde{t}_{k,m}$ of T^{-1} are given by

$$\tilde{t}_{k,m} = N^{-1} \alpha^{-km} \qquad k,m = 0, 1, \ldots N-1 \qquad (8.12)$$

This is proved by considering

$$TT^{-1} = I$$

or

$$N^{-1} \sum_{n=0}^{N-1} \alpha^{kn} \alpha^{-\ell n} = \delta_{k,\ell} \qquad (8.13)$$

where $\delta_{k,\ell}$ is the Kronecker delta function. Putting $k-\ell = p$ in equation (8.13) results in the requirement

$$N^{-1} \sum_{n=0}^{N-1} \alpha^{pn} = \begin{cases} 1 & \text{if } p = 0 \pmod N \\ 0 & \text{otherwise.} \end{cases} \qquad (8.14)$$

Thus to prove (8.12) requires that (8.14) be proved. If $p = 0 \pmod N$ then $\alpha^p = 1$ so that the first part is established. If $p \neq 0 \pmod N$, $\alpha^p \neq 1$ or $(\alpha^p - 1) \neq 0$. Multiplying equation (8.14) by $(\alpha^p - 1)$ gives

$$N^{-1} (\alpha^p - 1) \sum_{n=0}^{N-1} \alpha^{pn} = N^{-1} (\alpha^{pN} - 1) = 0 .$$

Thus (8.14) is established.

The preceding discussion shows that the existence of an $N \times N$ transform which has the cyclic convolution property depends only upon the existence of an α that is a root of unity of order N, and the existence of N^{-1}. It should be remembered that the latter condition does not necessarily hold for non-zero values of N if the transform is defined on fields other than the complex number field. The structures of the transform matrix T and its inverse are given by equations (8.11) and (8.12) respectively. These are the only structures which support the cyclic convolution property, and are jointly called the DFT structure. Because of this DFT structure, any transform which has the cyclic convolution property also has a fast computation algorithm similar to the FFT, although N may be restricted to highly composite values, for example integer powers of two. In the complex number field the DFT (with $\alpha = e^{j2\pi/N}$) is the only

transform which has the cyclic convolution property. However, in a finite field or, more generally, a ring, transforms with the cyclic convolution property exist provided a root of unity of order N exists and provided N^{-1} exists.

Agarwal and Burrus also investigate transforms which have the non-cyclic convolution property. Such transforms belong to a more general class than those with cyclic convolution, because equation (8.9) no longer follows from equation (8.8). Thus the $t_{k,1}$'s are not restricted to N^{th} roots of unity, and the only restriction is the non-singularity condition (which requires all the $t_{k,1}$'s to be distinct). Because of this, the FFT type fast algorithm is not generally applicable. Since non-cyclic convolution may be performed using cyclic convolution (by padding the sequences with zeros, cf. Section 7.4.2), this review considers only transforms with a cyclic convolution property.

8.3.1 Terminology

The number theoretic transforms whose structure has been considered above are implemented on a finite ring of integers, for which multiplication and addition are defined modulo an integer F. In such a ring commutativity, associativity and distributivity apply, but division is undefined (Rader, 1972b). Consequently some numbers other than zeros have no multiplicative inverse mod F, unless F is prime.

In a number theoretic transform an integer α which is a root of unity of order N replaces $e^{-j2\pi/N}$ which is

used in the DFT. When F is a Mersenne number ($F = 2^p - 1$, $p \in I^+$) the transform is called a Mersenne number transform (MNT). Similarly when F is a Fermat number ($F = 2^b + 1$ where $b = 2^t$, $t \in I^+$) the transform is called a Fermat number transform FNT. The particular MNT's and FNT's for which $\alpha = 2$ are called Rader transforms (RT's) (Agarwal and Burrus, 1974).

8.3.2 Computational Considerations

Agarwal and Burrus present a general theorem which gives the necessary and sufficient conditions which N and F must satisfy for a number theoretic transform to exist. For completeness this is reproduced here:

An $N \times N$ transform T having the cyclic convolution property in the ring of integers modulo an integer F exists if and only if N divides O(F) where

$$O(F) \triangleq \gcd(p_1-1, p_2-1, \ldots p_\ell-1) \; . \tag{8.15}$$

gcd here means greatest common divisor, and

$$F = p_1^{r1} \; p_2^{r2} \; \ldots \; p_\ell^{r\ell} \tag{8.16}$$

is the prime factorisation of F. The implications of this theorem are discussed by Agarwal and Burrus (1974). For a number theoretic transform to be computationally more attractive than the FFT, there are three requirements that must be satisfied. Firstly, N should be usefully large, and should be composite (e.g. a power of 2) so that a fast FFT type algorithm exists. Secondly, multiplication by powers of α should be a simple operation. For example, if powers of α are also a power of 2, then multiplication by

α reduces to a word shift which is faster and/or cheaper than conventional multiplication. Thirdly, for inexpensive arithmetic modulo F, F should be representable in few bits.

These considerations have led Agarwal and Burrus to conclude that FNT's are more useful than MNT's. This conclusion is based partly upon the criterion that N be an integer power of 2, which is not necessary if a mixed radix FFT type algorithm is used (Rader, 1968; Singleton, 1969). Reed and Truong (1975) show that MNT's can be used to convolve complex sequences, but state that this is not possible using FNT's. Complex convolution is discussed by Agarwal and Burrus (1975), who give a theorem analogous to that expressed in the paragraph containing equations (8.15) and (8.16).

Vegh and Leibowitz (1976) and Nussbaumer (1977) also consider the use of number-theoretic transforms for complex convolution. Further research is required to clarify the relative merits of FNT's and MNT's.

A major disadvantage of both FNT's and MNT's is that the word length required depends strongly upon N. Table 8.1, which is taken from Agarwal and Burrus (1974), lists various combinations of parameters for FNT implementations to illustrate this problem. Remember that for non-cyclic convolution the input sequences are padded with zeros, which compounds the problem further. However, as Rader (1972b) points out, this can be overcome by using a multi-dimensional transform. Table 8.2 (also from Agarwal and Burrus, 1974) shows how N can be increased while practical word lengths are retained.

The use of transforms other than the MNT and FNT are discussed by Lui, Reed and Truong (1976) and Nussbaumer (1977). Lui *et al.* consider rings modulo primes of the form $(2^m - 1)2^n + 1$. They show that high-radix FFT algorithms can be utilised to relax the restriction imposed by the register word length on the transform length. Nussbaumer (1977) considers "pseudo FNT's", which are applicable to rings modulo $2^b + 1$ where $b \neq 2^t$. This approach also permits greater flexibility in the choice of word lengths and transform lengths than is permitted using the FNT. In addition, "pseudo FNT's" may be used to perform complex convolution.

An FNT algorithm has been implemented in ASSEMBLY language for an IBM 360/155 by Agarwal and Burrus (1974), and compared with an efficient FFT algorithm. This shows that the FNT is faster than the FFT by a factor of 3 to 5 for $N \leq 256$. For values of N greater than this the speed advantage is reduced to about 2. Presumably if the hardware were designed specifically for modulo F arithmetic then the speed advantage would be even greater.

A special purpose pipelined hardware implementation of a 64 point, 16 bit FNT is described by McClellan (1976), who also considers the hardware requirements for a 1024 point real aperiodic convolver, and compares the pipelined FFT and FNT methods. This analysis shows that the FNT requires more memory and more computational elements (i.e. "butterflies") than does the FFT. However, the FNT method can result in significant hardware cost savings because the FNT "butterflies" are much cheaper than the FFT "butterflies"

(typically the former is only 15% to 30% of the cost of the latter). This conclusion applies only for real convolutions. In addition, as the transform length is increased the cost advantage of the FNT convolver over the FFT convolver must decrease, because the memory cost of a pipeline processor increases faster than the "butterfly" cost.

8.4 WALSH TRANSFORM METHODS

The preceding sections of this chapter have considered the efficient computation of convolutions using the Fourier transform, or using number theoretic transforms which are analogous to the Fourier transform. In this section are discussed methods based on the Walsh transform, which is also called the Walsh-Fourier transform, Walsh-Hadamard transform and Hadamard transform.

8.4.1 The Walsh Transform

It is well known that the trigonometric (or complex exponential) functions form a complete orthogonal set, and that this orthogonality is the basis of the Fourier transform (cf. Papoulis, 1962; Lathi, 1965). Numerous other functions also form a complete orthogonal set.

Of particular interest are the Walsh and Haar sets of functions. These assume, respectively, only two and three values, and are consequently well matched to digital techniques and semiconductor technology. In addition, the Walsh and Haar transforms exhibit a smaller computation-speed / storage-space product than the corresponding Fourier

transform. This latter property provided the initial motivation for the author's investigation of these transforms.

Detailed discussion of both the Walsh and Haar functions and their applications are given by Harmuth (1972, 1977) and Beauchamp (1975). While these provide a comprehensive background, they give scant consideration to the use of the transforms in computing conventional convolutions.

The Haar transform is the fastest linear transform known at present (cf. Beauchamp, 1975) and is simpler to implement than the Walsh transform. However, it does not seem to be as useful as the latter and is not considered here. It is also worth noting that the Walsh and Haar transforms have been generalised. A useful discussion of such generalised transforms is given by Elliott (1974).

The use of the Walsh transform in the Hapstrum technique of pitch estimation has already been discussed (see Section 7.5.5).

Walsh functions form an ordered set of rectangular waveforms which take only two amplitude values + 1 and - 1. This set is complete and orthogonal, and is defined over a finite interval T which is called the time base. The continuous Walsh function of order n is written wal(n,t). The time period t is usually normalised to t/T . Two major ordering conventions are in use - these are "sequency order" (also called ordered form, Walsh order, Walsh-Kaczmarz order) and "natural order" (also called normal order, binary order, dyadic order, Paley order). Sequency order

corresponds to arranging the Walsh functions so that the number of zero crossings within the time base increases with n. This form is useful in spectral analysis applications, because of the analogy between sequency and frequency. Natural ordering may be achieved from sequency ordering by Gray Code transformation of n (cf. Yuen, 1971). Natural order arises when the Walsh functions are defined as products of Rademacher functions (which correspond to hard-limited sinusoids, and themselves form an orthogonal but incomplete set (Harmuth, 1972)). Natural order also has analytic advantages over sequency order.

The discrete Walsh functions are sampled versions of the continuous set, for which numerous definitions exist. The complete set of N-length discrete Walsh functions is obtained by an N-point sample of the continuous functions on the interval [0, 1], and for convenience N is herein constrained to be an integral power of 2. The sequence is defined to be zero outside the interval [0, 1].

A convenient sequency ordered definition of the N-length discrete Walsh functions is given by Kennett (1970):

$$\mathrm{wal}(k,j) = \prod_{r=0}^{p-1} (-1)^{(k_{p-r} + k_{p-r-1})j_r} \qquad (8.17)$$

$$j = 0, 1, 2, \ldots N-1$$

$$k = 0, 1, 2, \ldots N-1$$

where $N = 2^p$ and j_r, k_r here denote the bits of the binary representation of j,k (viz. $j = \sum_{r=0}^{p-1} j_r 2^r$). This definition is illustrated for $N = 8$ by expressing $\mathrm{wal}(k,j)$

as an 8×8 matrix:

$$\text{wal}(k,j) = \begin{array}{c} j = \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & - & - & - & - \\ 1 & 1 & - & - & - & - & 1 & 1 \\ 1 & 1 & - & - & 1 & 1 & - & - \\ 1 & - & - & 1 & 1 & - & - & 1 \\ 1 & - & - & 1 & - & 1 & 1 & - \\ 1 & - & 1 & - & - & 1 & - & 1 \\ 1 & - & 1 & - & 1 & - & 1 & - \end{bmatrix} \end{array} \begin{array}{c} k = \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \qquad (8.18)$$

where - denotes -1.

From this representation of the discrete Walsh functions the following properties are easily deduced. They are orthogonal:

$$\sum_{j=0}^{N-1} \text{wal}(k,j)\ \text{wal}(\ell,j) = \begin{cases} N & k = \ell \\ 0 & k \neq \ell \end{cases} . \qquad (8.19)$$

Also they are symmetric:

$$\text{wal}(k,j) = \text{wal}(j,k) . \qquad (8.20)$$

There is an addition formula:

$$\text{wal}(\ell,j)\ \text{wal}(k,j) = \text{wal}(\ell \oplus k, j) \qquad (8.21)$$

where $\oplus$ indicates addition modulo 2 (i.e., bit by bit addition with no carry, viz. $0 \oplus 1 = 1 \oplus 0 = 1$, $1 \oplus 1 = 0 \oplus 0 = 0$). This is also the exclusive OR function, XOR.

For an N-length real sequence $\{f_j\}$ the finite Walsh transform is defined as

$$F_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \, wal(k,j) \qquad k = 0, 1, 2, \ldots N-1 . \quad (8.22)$$

Similarly f_j is expressed as the inverse finite Walsh transform; thus

$$F_j = \sum_{k=0}^{N-1} F_k \, wal(k,j) \qquad j = 0, 1, 2, \ldots N-1 . \quad (8.23)$$

$$= \sum_{k=0}^{N-1} F_k \, wal(j,k)$$

Fast Walsh transforms exist for both sequency and natural ordering (cf. Beauchamp, 1975). Denote Walsh transform pairs as

$$\{f_j\} \overset{WT}{\leftrightarrow} \{F_k\} .$$

Then the following properties apply (cf. Kennett, 1970).

(i) The finite Walsh transform is linear. If

$$\{f_j\} \overset{WT}{\leftrightarrow} \{F_k\}$$

and

$$\{g_j\} \overset{WT}{\leftrightarrow} \{G_k\}$$

then

$$a\{f_j\} + b\{g_j\} \overset{WT}{\leftrightarrow} a\{F_k\} + b\{G_k\} \quad (8.24)$$

for real constants a and b .

(ii) Symmetry. wal(k,j) is symmetric about the midpoint of the sequence j = 0, 1, 2, ... N-1 if k is even, and asymmetric if k is odd. Thus a sequence $\{f_j\}$ which is

symmetric about its midpoint will have a transform $\{F_k\}$ which is composed solely of even-order Walsh function coefficients. Similarly if $\{f_j\}$ is asymmetric about its midpoint then $\{F_k\}$ consists only of odd-order coefficients.

(iii) A delta function $\delta(n)$ exists such that

$$\{\delta_j\} \overset{WT}{\leftrightarrow} 1/N$$

$$1 \overset{WT}{\leftrightarrow} \{\delta_k\} \quad . \tag{8.25}$$

The sequence δ_n is defined by $\delta_n = 1$ if $n = 0$, $\delta_n = 0$ otherwise. This is useful in expressing the transform of a shifted sequence:

$$\{f_j\} - a \overset{WT}{\leftrightarrow} \{F_k\} - a\{\delta_k\} \tag{8.26}$$

(iv) If $\{f_j\}$ is a set of observations, F_0 is the sample mean.

$$f_0 = \sum_{k=0}^{N-1} F_k$$

$$F_0 = \frac{1}{N} \sum_{k=0}^{N-1} f_j \quad . \tag{8.27}$$

(v) Convolution. Consider the convolution of two N-length sequences $\{f_j\}$, $\{g_j\}$ such that $\{f_j\} \overset{WT}{\leftrightarrow} \{F_k\}$, $\{g_j\} \overset{WT}{\leftrightarrow} \{G_k\}$. The first N values of the convolved sequence $\{h\} = \{f\} * \{g\}$ is obtained from

$$h_s = \frac{1}{N} \sum_{r=0}^{N-1} f_r g_{s-r} \tag{8.28}$$

where $g_m = 0$ if $m < 0$, and $s = 0, 1, 2, \ldots N-1$.

The factor 1/N is introduced here (cf. equation 8.2) following Kennett's definition, because this results in a direct analogy between the conventional and logical convolution (cf. equations 8.31 to 8.33 below). Thus

$$h_s = \frac{1}{N} \sum_{r=0}^{N-1} \left[\sum_{k=0}^{N-1} F_k \, \mathrm{wal}(k,r) \right] \cdot \left[\sum_{\ell=0}^{N-1} G_k \, \mathrm{wal}(\ell, s-r) \right]$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} F_k G_\ell \left[\sum_{r=0}^{N-1} \mathrm{wal}(k,r) \, \mathrm{wal}(\ell, s-r) \right] \qquad (8.29)$$

where the expression in brackets is the convolution of the discrete Walsh functions (note that $\mathrm{wal}(k,m) \equiv 0$ for $m < 0$). An alternative expression for h_s may also be derived:

$$h_s = \frac{1}{N} \sum_{r=0}^{N-1} f_r \left[\sum_{k=0}^{N-1} G_k \, \mathrm{wal}(k, s-r) \right]$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} G_k \left[\sum_{r=0}^{N-1} f_r \, \mathrm{wal}(k, s-r) \right] . \qquad (8.30)$$

The complexity of these expressions arises from the lack of a simple arithmetic shifting relation for the Walsh functions.

(vi) "Logical convolution" (Gibbs, 1967) is a function of two sequences $\{f_r\}$ and $\{g_r\}$ which is the Walsh transform analogy of Fourier transform convolution. The logical convolution is defined as

$$h_s = \frac{1}{N} \sum_{r=0}^{N-1} f_r g_{s \oplus r} \tag{8.31}$$

where $\oplus$ indicates addition modulo 2 (i.e. "logical addition"). The symbolic representation of logical convolution is

$$\{h\} = \{f\} \circledast \{g\} \tag{8.32}$$

It is easily proved (Kennett, 1970) that

$$\{f_j\} \circledast \{g_j\} \overset{WT}{\leftrightarrow} \{F_k G_k\} \tag{8.33}$$

(vii) Logical convolution leads to an analogue of the Wiener-Khintchin theorem (cf. equation 8.1). Define the logical autocorrelation of a N-length sequence $\{f_j\}$, where $\{f_j\} \overset{WT}{\leftrightarrow} \{F_k\}$, as

$$\{L_j\} = \{f_j\} \circledast \{f_j\}$$

$$= \frac{1}{N} \sum_{m=0}^{N-1} f_{j \oplus m} f_m \tag{8.34}$$

and the Walsh power spectrum $\{P_k\}$ by the relation

$$\{P_k\} = \{F_k\}^2 \tag{8.35}$$

But from equation (8.33)

$$\{f_j\} \circledast \{f_j\} \overset{WT}{\leftrightarrow} \{F_k{}^2\} \tag{8.36}$$

so that

$$\{L_j\} \overset{WT}{\leftrightarrow} \{P_k\} \tag{8.37}$$

which is the analogue of the Wiener-Khintchin theorem.

(viii) The Walsh function analogue of Parseval's theorem is

$$\frac{1}{N} \sum_{j=0}^{N-1} f_j^2 = \sum_{k=0}^{N-1} F_k^2 \quad . \tag{8.38}$$

The proof is outlined by Kennett (1970).

The preceding summary is necessarily brief, but it introduces the definitions necessary for the discussion which follows. It is now advantageous to compare the FFT and the FWT.

Since the Walsh functions take only the values ±1, the FWT is faster and much simpler to implement than the corresponding FFT. This simplification arises because the complex multiplications required by the FFT are replaced by multiplication by ±1 in the FWT. Consequently the sum of products of the FFT can be implemented using only addition and subtraction in the FWT. Geadah and Corinthios (1977) present algorithms for the efficient computation of the FWT for natural, dyadic and sequency order.

Beauchamp (1975) gives a table of comparative computation times, which indicates that the FWT is typically 4 to 6 times faster than the FFT. However, Landwehr (1973) gives a more comprehensive table which lists the computation times for a variety of computers. Landwehr's results indicate that the speed advantage of the FWT is typically 1.5 to 4. This speed advantage of the FWT over the FFT can be expected to decrease as advances in integrated circuit technology produce faster multipliers.

The lack of a simple expression for circularly shifting the Walsh functions has already been noted (cf. equations 8.29 and 8.30). The main consequences of this are that the Walsh coefficients in the DWT are phase-dependent, and that arithmetic (i.e. conventional) convolution cannot be performed simply. The phase dependence of the DWT is illustrated in Figure 8.1. Several approaches are now considered which aim to overcome this deficiency, so that the speed and simplicity of the FWT can be utilised. These approaches include developing Walsh-like transforms which are invariant to cyclic shifts, and developing a fast transformation which coverts a logical convolution into an arithmetic (i.e., conventional) convolution.

8.4.2 Walsh-Related Transforms with Cyclic Shift Invariance

The dyadic rather than cyclic shift invariance of the Walsh transform prompted the development of several Walsh-related transforms which are invariant to cyclic shifts. The earliest of these appears to be the R-transform, which is described by Reitboeck and Brody (1968). Ulman (1970) shows that the only computational difference between the Walsh and R-transforms is that all subtraction operations (a-b) in the former are replaced by |a-b| in the latter. Unfortunately this non-linearity makes the R-transform non-invertible. It also results in a substantial reduction in the information about the sequency distribution of the transformed coefficients, particularly in the high sequency range (Beauchamp, 1975). This is

illustrated in Figure 8.2. The non-invertibility of the R-transform makes it unsuitable for computing convolutions.

Ohnsorg (1971) derives a transform which operates on the DWT coefficients to produce a spectrum which is invariant to cyclic shifts. This transform takes the form of a combination of quadratic and paired-product terms of the DWT coefficients, and is called the WT quadratic spectrum or Q spectrum (Ahmed, Abdussattar and Rao, 1972). The derivation and definition of the Q-spectrum is given by Ohnsorg (1971) and Ahmed *et al.* (1972) and is not repeated here. It suffices to illustrate the approach used by giving an example for the case $N = 8$, where N is the length of the sequence $\{x_j\} \overset{WT}{\leftrightarrow} \{X_k\}$. Then

$$
\begin{aligned}
Q_0 &= X_0^2 \\
Q_{1,0} &= X_1^2 \\
Q_{2,0} &= X_2^2 + X_3^2 \\
Q_{3,0} &= X_4^2 + X_5^2 + X_6^2 + X_7^2 \\
Q_{3,1} &= \frac{1}{2}\{X_4^2 - X_5^2 + X_6^2 - X_7^2 - 2X_4X_7 + 2X_5X_6\} \qquad (8.39)
\end{aligned}
$$

where $Q_{m,q}$ are the Q-spectrum coefficients. In general there are $N/2 + 1$ coefficients of the Q-spectrum, so that the spectral resolution is significantly reduced.

Ohnsorg (1971) refines the Q-spectrum to obtain the optimum quadratic spectrum, which is called the J-spectrum by Ahmed *et al.* (1972). The optimum quadratic spectrum is obtained by a further transformation of the Q-spectrum, and is so called because it optimises the spectral resolution

which is attainable by the quadratic spectrum. This is achieved by rearranging the squared DWT coefficients so that "scrambling" of the spectral sequencies is minimised. This is illustrated by the J-spectrum corresponding to the example of equation (8.39):

$$
\begin{aligned}
J_0 &= X_0^2 \\
J_1 &= X_1^2 \\
J_2 &= X_2^2 + X_3^2 \\
J_{3,0} &= X_4^2 + X_6^2 - X_4X_7 + X_5X_6 \\
J_{3,1} &= X_5^2 + X_7^2 + X_4X_7 - X_5X_6 \; .
\end{aligned} \tag{8.40}
$$

Ohnsorg (1971) notes that the approach used to derive the cyclic shift-invariant transforms can also be used to derive analogues of the circular convolution and correlation theorems. There is little point in so doing because of the amount of computation required to evaluate the J- and Q-spectra. Ahmed *et al.* (1972) give fast algorithms for the matrix transformations derived by Ohnsorg. Their implementation (on an IBM 360/50) required 0.5 minutes and 1.6 minutes to compute the Q- and J-spectra, respectively, for a 1024 point sequence. These times are greater than the cyclic convolution of a 1024 point sequence using the FFT on a similar machine (cf. Table III of Agarwal and Burrus, 1974). For this reason there has been little subsequent investigation of cyclic shift-invariant relatives of the DWT. Instead, most research in this field has been directed toward the number-theoretic transforms which are discussed

in Section 8.3.

8.4.3 Autocorrelation from Logical (Dyadic) Correlation

The analogy between the logical (i.e. dyadic) and arithmetic (i.e. conventional) Wiener-Khintchin theorems is noted in equation (8.37). Since logical correlations can be computed faster and/or more cheaply than arithmetic correlations, it is worth investigating whether a transformation exists which generates the latter from the former. If such a transform exists, then its complexity and speed will determine the usefulness of this approach.

Denote by T_{A-L} the transformation from the arithmetic to logical autocorrelation function of a sequence {f}, and by T_{L-A} the transformation from logical to arithmetic autocorrelation function of {f} . Gibbs (1967) deduced that if the input sequence {f} is real and representative of a discrete, wide-sense stationary, stochastic process, then T_{A-L} and T_{L-A} exist as linear transforms. This is proved by Pichler (1970), and discussed further by Gibbs and Pichler (1971). Robinson (1972) derives this transformation in matrix form by comparing dyadic shifts and time shifts. She also shows that the symmetry of the transform matrix permits the relationship between the logical and arithmetic autocorrelation functions to be expressed recursively. Lopresti and Suri (1974) factor the transformatin matrix and present a fast algorithm. They show that the computation of the arithmetic autocorrelation function using the series of transforms

$$\{f\} \overset{WT}{\leftrightarrow} \{F_W\} \overset{(\,)^2}{\leftrightarrow} \{P_W\} \overset{WT^{-1}}{\leftrightarrow} \{L\} \overset{T_{L-A}}{\leftrightarrow} \{\phi\} \qquad (8.41)$$

is substantially more efficient than the conventional frequency domain method

$$\{f\} \overset{FT}{\leftrightarrow} \{F_F\} \overset{||^2}{\leftrightarrow} \{P_F\} \overset{FT^{-1}}{\leftrightarrow} \{\phi\} \qquad (8.42)$$

where $\{f\}$ denotes the input sequence, $\{F_F\}$ and $\{F_W\}$ denote the Fourier and Walsh transforms of $\{f\}$, $\{P_F\}$ and $\{P_W\}$ denote the Fourier and Walsh power spectra, and $\{L\}$ and $\{\phi\}$ denote the logical and arithmetic autocorrelation functions, respectively. For example, for a 1024 point real sequence $\{f\}$, the FFT method (relation 8.42) requires a total of 94,208 multiplications and 94,208 additions. In contrast, the FWT method (relation 8.41) requires only 3072 multiplications and 25,601 additions. This can be further simplified, since 1024 of the multiplications involve multiplication by an integer power of 2 and can therefore be implemented using word shifts.

The approach outlined above is applicable only if the input sequence $\{f\}$ is a wide-sense stationary stochastic process, and little work seems to have been done on generalising the method to other functions, such as the quasi-periodic signals encountered in pitch estimation. However, the requirement of wide-sense stationarity is used explicitly in the derivations and proofs given by the workers cited above. The blind use of the method expressed by relation (8.41) yields results of the form illustrated in Figure 8.3(d), which shows the autocorrelation function

of a segment of voiced speech. Comparison of Figures 8.3(d) with Figure 8.3(c) (which is the directly-computed autocorrelation function of the same signal) shows that this method is not suitable for pitch estimation.

8.4.4 Cyclic Convolution from the Walsh Transform

Pitassi (1971) describes an efficient algorithm for computing cyclic convolutions of sample sequences of length $N = 2^M$ where N,M are integers. Pitassi's method is essentially a factorisation of the convolution equation (cf. equation 8.28). Each iteration of his algorithm requires three sub-convolutions of sequences whose lengths are half the input sequence length at the current iteration level. In this way the cyclic convolution is performed using $2(3^{M-1})$ multiplications, which is more efficient than the FFT method for values of M less than 10 (i.e. for sequences of length N less than 1024).

Pitassi's approach is summarised below. Consider the circular convolution $\{r\} = \{x\} * \{y\}$, which is defined as

$$r_n = \sum_{m=0}^{N-1} x_m y_{m+n} \qquad n = 0, 1, \ldots N-1 \qquad (8.43)$$

where the sequences {r}, {x} and {y} are of length N and the indices of equation (8.43) are evaluated modulo N. Note that this definition is strictly that of a circular correlation (cf. equation 8.2). However, bearing in mind the comments in the second paragraph of Section 8.1, Pitassi's definition is retained here. Constrain N to be an integer power of 2, say $N = 2^M$. Then equation (8.43) may

be rewritten:

$$r_n = \sum_{m=0}^{\frac{N}{2}-1} x_{2m}\, y_{2m+n} + \sum_{m=0}^{\frac{N}{2}-1} x_{2m+1}\, y_{2m+1-n} \,. \qquad (8.44)$$

The odd and even components of {r} can be written as

$$r_{2n} = \sum_{m=0}^{\frac{N}{2}-1} x_{2m}\, y_{2m+2n} + \sum_{m=0}^{\frac{N}{2}-1} x_{2m+1}\, y_{2m+2n+1} \qquad (8.45)$$

and

$$r_{2n+1} = \sum_{m=0}^{\frac{N}{2}-1} x_{2m}\, y_{2m+2n+1} + \sum_{m=0}^{\frac{N}{2}-1} x_{2m+1}\, y_{2m+2n+2} \,. \qquad (8.46)$$

Define operators E{x} and O{x} such that

$$E\{x\} = \begin{bmatrix} x_0 \\ x_2 \\ x_4 \\ \cdot \\ \cdot \\ \cdot \\ x_{N-2} \end{bmatrix} \qquad O\{x\} = \begin{bmatrix} x_1 \\ x_3 \\ x_5 \\ \cdot \\ \cdot \\ \cdot \\ x_{N-1} \end{bmatrix} \qquad (8.47)$$

where

$$\{x\} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{N-1} \end{bmatrix} \quad \text{and} \quad \{x\}' = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ \cdot \\ x_{N-1} \\ x_0 \end{bmatrix}$$

Then equations (8.45) and (8.46) can be expressed as

$$E\{r\} = E\{x\} * E\{y\} + O\{x\} * O\{y\} \tag{8.48}$$

$$O\{r\} = E\{x\} * O\{y\} + O\{x\} * E\{y\}' \tag{8.49}$$

Now define operators P{x} and S{x} such that

$$P\{x\} = \begin{bmatrix} x_0+x_1 \\ x_2+x_3 \\ . \\ . \\ . \\ x_{N-2}+x_{N-1} \end{bmatrix} \qquad S\{x\} = \begin{bmatrix} x_0-x_1 \\ x_2-x_3 \\ . \\ . \\ . \\ x_{N-2}x_{N-1} \end{bmatrix} \tag{8.50}$$

and note that $P\{x\} = E\{x\} + O\{x\}$, $S\{x\} = E\{x\} - O\{x\}$. Consider the auxiliary convolution functions {c} and {d}, defined as

$$\{c\} = P\{x\} * P\{y\} = (E+O)\{x\} * (E+O)\{y\} \tag{8.51}$$

$$\{d\} = S\{x\} * S\{y\} = (E-O)\{x\} * (E-O)\{y\} \tag{8.52}$$

Adding (8.51) and (8.52) yields

$$\{c\} + \{d\} = 2(E\{x\} * E\{y\} + O\{x\} * O\{y\}) . \tag{8.53}$$

Similarly,

$$\{c\} - \{d\} = 2(E\{x\} * O\{y\} + O\{x\} * E\{y\}) . \tag{8.54}$$

Comparison of equations (8.48) with (8.53) shows that

$$\{c\} + \{d\} = 2\ E\{r\} . \tag{8.55}$$

However, the same is not true of equations (8.49) and (8.54) because of the circular shift of the second term of the right-hand side of (8.49). A third auxiliary convolution is introduced to correct equation (8.54) so that the odd indexed components of {r} are produced. This auxiliary convolution is denoted by {f} and defined as

$$\{f\} = (P - S)\{x\} * E\{y\} = 2\,O\{x\} * E\{y\} . \qquad (8.56)$$

Since

$$\{x\} * \{y\}' = (\{x\} * \{y\})' \qquad (8.57)$$

the desired correction term is $\{f\}' - \{f\}$. Adding this correction term to equation (8.54) yields

$$\{c\} - \{d\} + \{f\}' - \{f\} = 2(E\{x\} * O\{y\} + O\{x\} * E\{y\}')$$

$$= 2\,O\{r\} . \qquad (8.58)$$

The preceding discussion shows that the desired convolution {r} can be obtained by linear combination of the three auxiliary convolutions {c}, {d} and {f}.

Pitassi provides algorithms for computing {r} in this manner. The signal flow diagram for part of this algorithm is identical to that of the Walsh transform algorithm given by Shanks (1969). Thus, the input sequences at each level of iteration are essentially Walsh-transformed, "shuffled", and re-combined to form the input sequences to the next level of iteration.

This approach to cyclic convolution is generalised by Davis (1972), who shows that Pitassi's method is a special case of a general class of methods. Davis derives efficient

algorithms for both cyclic and linear (i.e., aperiodic or non-cyclic) convolution. He also notes that linear convolution can be performed faster using his linear convolution algorithm than using cyclic convolution, because the latter requires that the input sequence be padded with zeros. In addition he shows that partial convolution (i.e., a segment of the complete convolution) may be computed efficiently using this approach.

In generalising this approach Davis derives two distinct methods, which he calls "halving by parisection" and "halving by bisection". These two methods are similar to the two approaches used in deriving the FFT algorithm (viz. decimation-in-time and decimation-in-frequency, respectively). Davis also clarifies the relationship between these methods and the Walsh transform.

To the author's knowledge, no name has been given to these methods of convolution. It is worth noting that Agarwal and Burrus (1974) suggest that Pitassi's method is a suitable adjunct to the FNT, so that long sequences may be efficiently convolved using a two-dimensional convolution.

8.5 CONCLUSION

In this chapter are reviewed a variety of transform methods for computing both cyclic and linear convolutions. These methods include the familiar FFT, number theoretic transforms - in particular the FNT, MNT and RT - and several techniques which are based on Walsh transforms. Of the latter, the fast transformation T_{L-A} (which converts the logical autocorrelation function into the arithmetic

autocorrelation function) is not suitable for pitch estimation, because it requires that the input signal be a stationary stochastic process. This is unfortunate, since for such signals the method is very much more efficient than conventional computation using the FFT (cf. Section 8.4.3). The attempts to derive Walsh-related transforms which are phase insensitive and possess conventional convolution theorems are essentially unsuccessful. Those transforms which are phase insensitive (cf. the Q- and J-spectra, Section 8.4.2) are comparatively complicated to compute, and the potential advantage offered by the FWT is lost. The most suitable Walsh-transform method is that of Pitassi (1971) (cf. Section 8.4.4) which is generalised by Davis (1972). These algorithms permit both cyclic and linear convolutions to be computed more efficiently than by the FFT method, provided the sequence length is small (less than about 1024 samples). Even so the margin of superiority is not large.

Number theoretic transforms are discussed in Section 8.3. These permit cyclic convolutions to be computed more efficiently than when the FFT is used. When implemented on general purpose computers the speed superiority over the FFT method is of the order of 2 for sequence lengths of the order of 256 to 2048. This would presumably be improved if the arithmetic were performed using hardware designed to perform modulo F arithmetic, where F is an integer appropriate to the particular transform being used. The main advantage of these number theoretic transforms is that they eliminate the truncation

and roundoff errors which accrue with the FFT.

None of the transform methods reviewed here offers substantial advantages over the FFT method of computing convolutions and correlations, unless the sequence lengths are very small. It is worth noting here that recently a variety of high speed programmable signal processors have been developed. Freeny (1975) discusses the hardware components of such processors, while Allen (1975) and Peled (1976) consider their design and architecture. Allen also describes a number of such machines which compute a 1024 point complex FFT in 5 ms to 8.5 ms. Consequently it appears very likely that the well established FFT will remain the most useful transform for computing convolutions and correlations.

In the light of this conclusion, the comments in Chapter 7 (cf. Sections 7.4.1 and 7.4.6) which pertain to pitch estimation using the autocorrelation function need not be modified.

TABLE 8.1

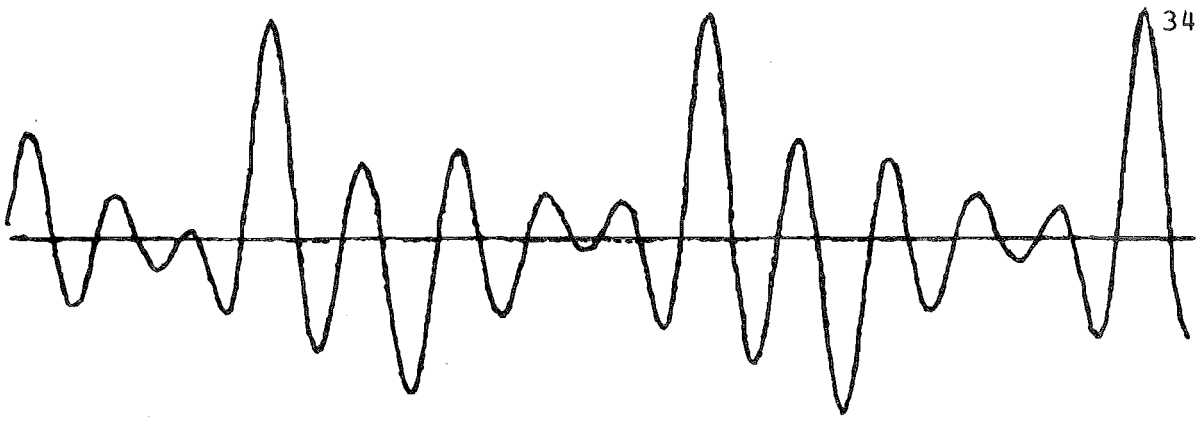
Parameters for several possible implementations of FNT's. (From Agarwal and Burrus, 1974).

t	b = No. of bits	$F_t = 2^{2^t}$	$\alpha^N = 2$	$\alpha^N = \sqrt{2}$	N_{max}	α for N_{max}
3	8	$2^8 + 1$	16	32	256	3
4	16	$2^{16} + 1$	32	64	65 536	3
5	32	$2^{32} + 1$	64	128	128	$\sqrt{2}$
6	64	$2^{64} + 1$	128	256	256	$\sqrt{2}$

TABLE 8.2

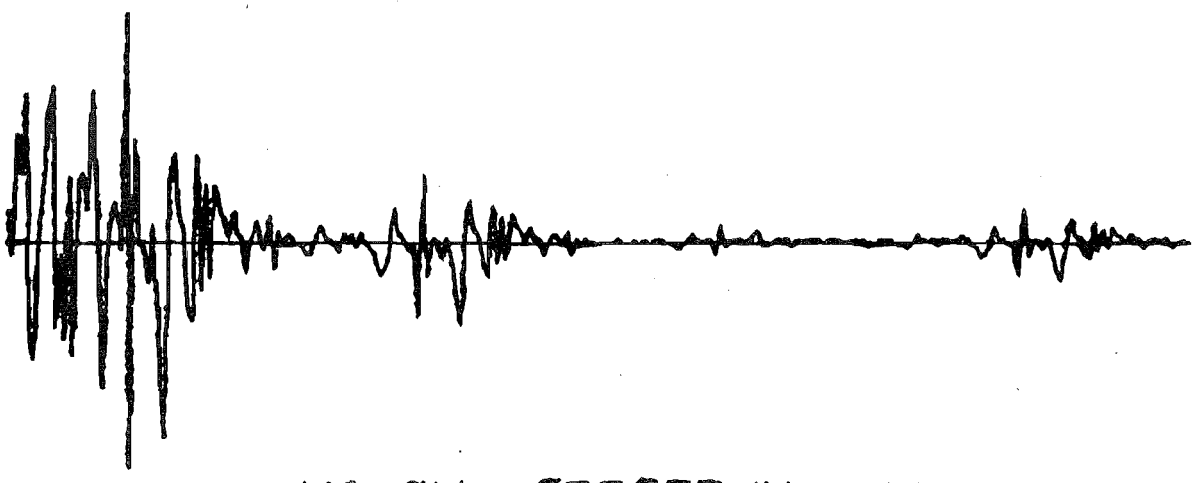
Maximum one-dimensional cyclic convolution lengths using two-dimensional FNT or RT. (From Agarwal and Burrus, 1974).

Word length b	N for $\alpha = 2$	N for $\alpha = \sqrt{2}$
16	512	2 048
32	2 048	8 192
64	8 192	32 768



ANALYSED SIGNAL. (a)

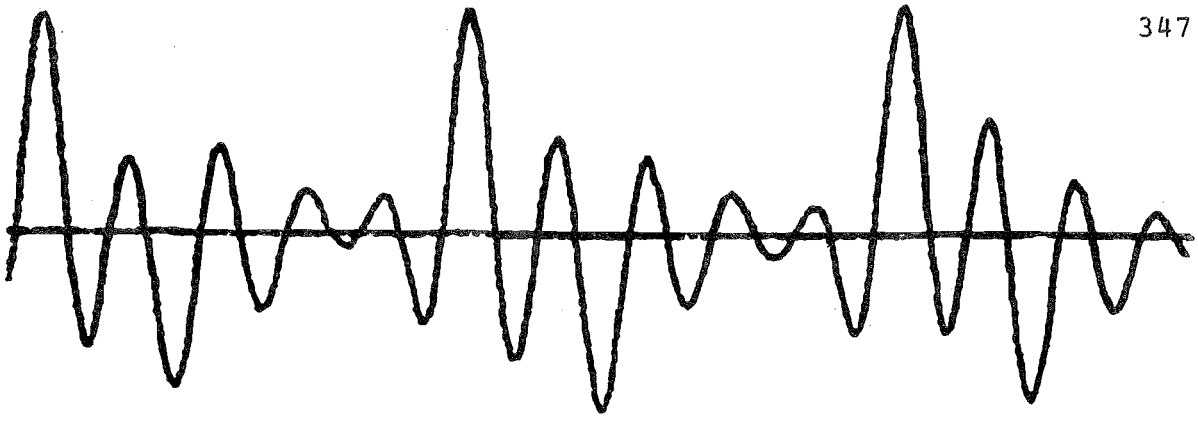
FOURIER SPECTRUM. (b)



WALSH SPECTRUM. (c)

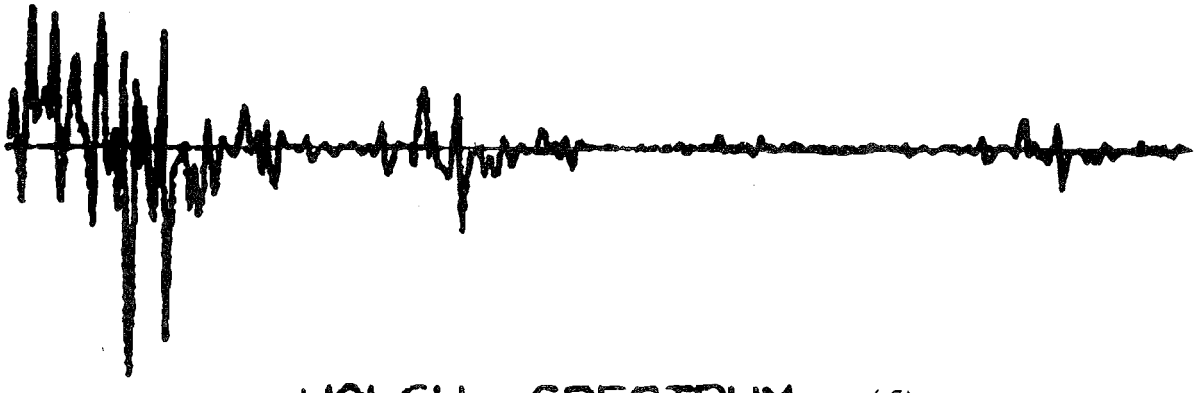
Figure 8.1 Illustrating the effect of cyclic shift on the Fourier and Walsh spectra.

(a) Speech signal segment
(b) Discrete Fourier transform of the segment (a)
(c) Discrete Walsh transform of the segment (a)



ANALYSED SIGNAL. (d)

FOURIER SPECTRUM. (e)



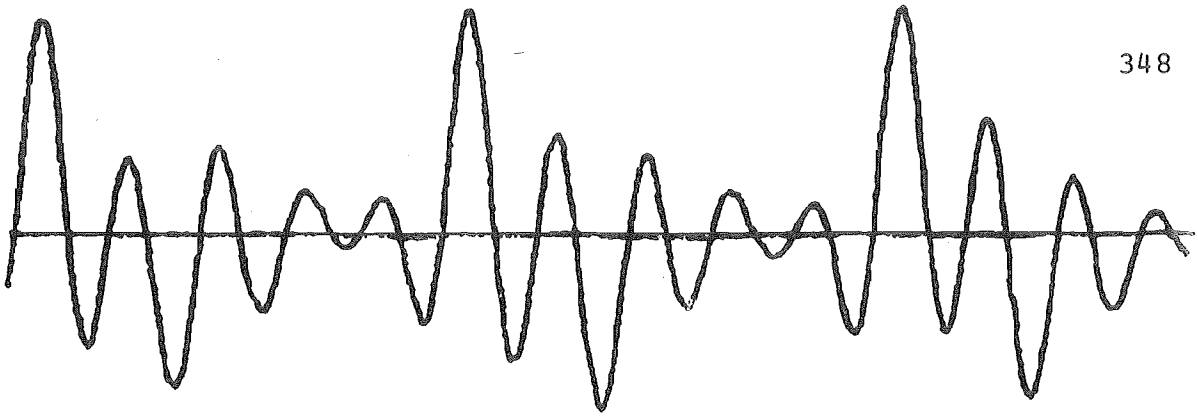
WALSH SPECTRUM. (f)

Figure 8.1 (Continued).

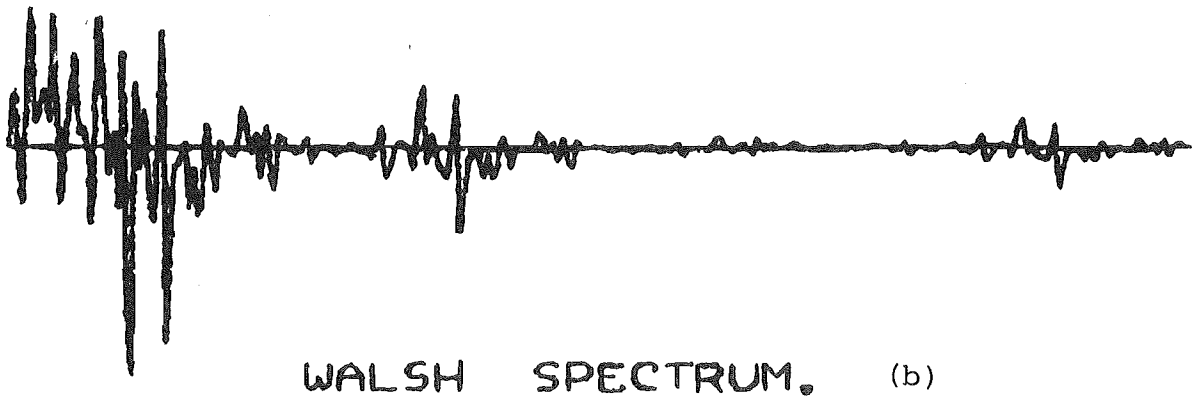
(d) The signal segment shown in (a) after displacement in time by approximately half a pitch period

(e) The discrete Fourier transform of the segment (d)

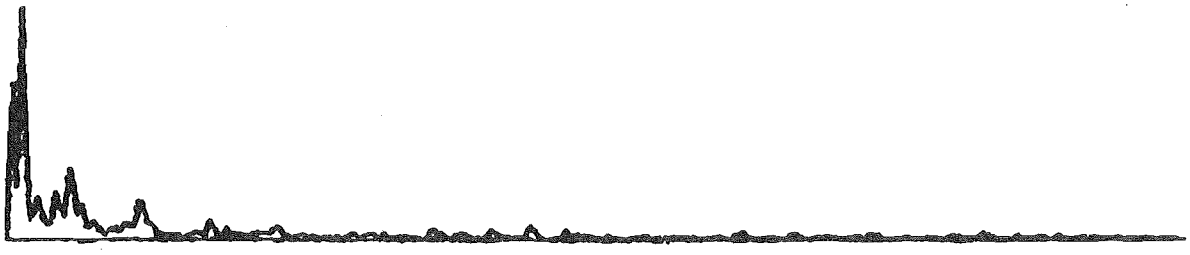
(f) The discrete Walsh transform of the segment (d).



ANALYSED SIGNAL. (a)



WALSH SPECTRUM. (b)



R SPECTRUM. (c)

Figure 8.2 Comparing the Walsh transform with the R-transform.

(a) Speech signal segment
(b) The discrete Walsh transform of the segment (a)
(c) The discrete R-transform of the segment (a).

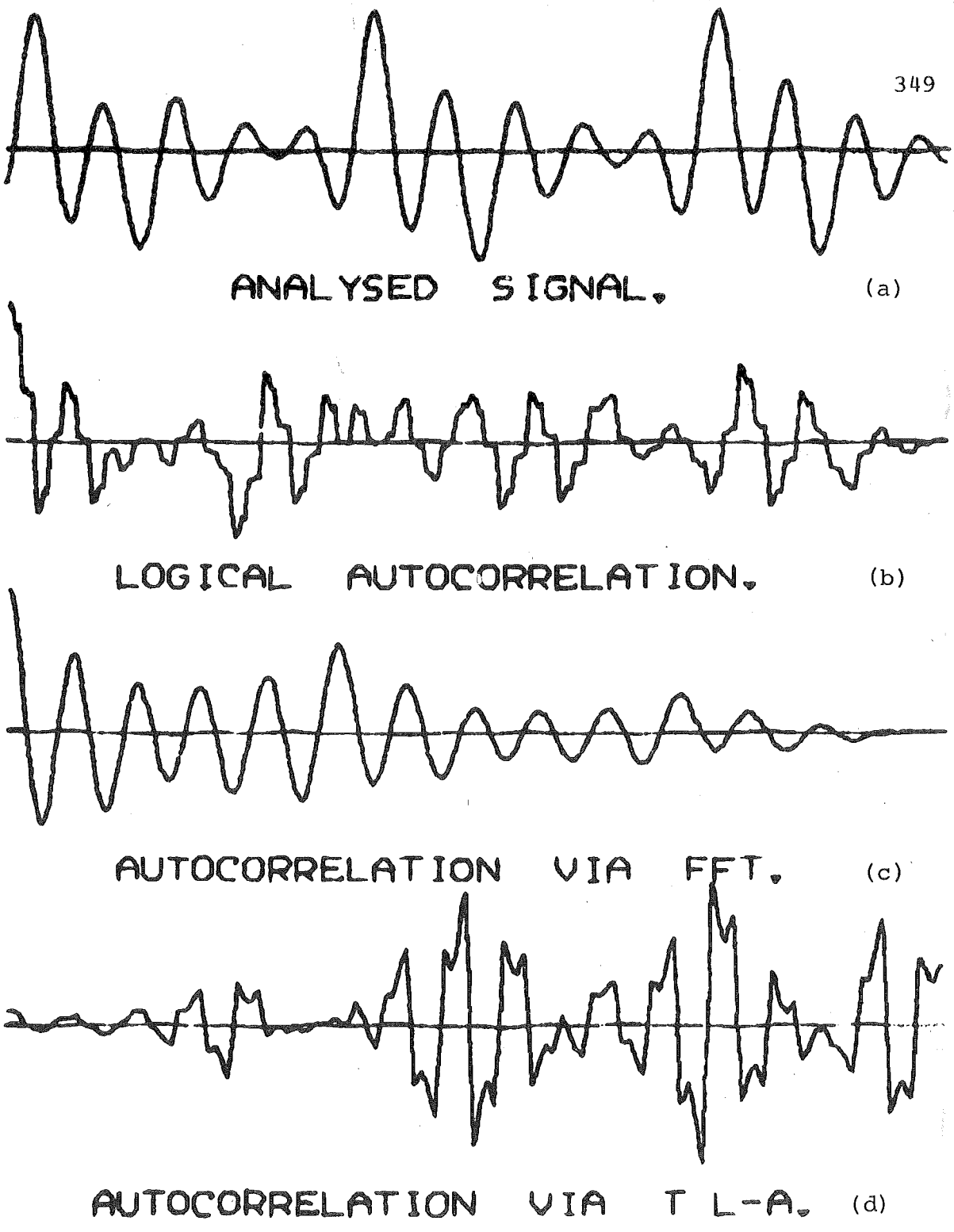


Figure 8.3 Comparing the logical (dyadic) and arithmetic autocorrelation functions.

(a) Speech segment
(b) The logical (dyadic) autocorrelation of (a)
(c) The arithmetic autocorrelation of (a)
(d) The arithmetic autocorrelation of (a) as obtained from (b) using the transform T_{L-A}.

CHAPTER 9

PITCH ESTIMATION SYSTEMS FOR SPEECH AND MUSIC

9.1 PITCH ESTIMATION USING TIME DOMAIN FEATURE RECOGNITION

Of the numerous pitch estimation techniques which are reviewed in Chapter 7, the only methods which are suitable for fast, inexpensive operation over wide pitch trajectory bandwidths seem to be those which operate in the time domain by recognising recurring features of the signal. This conclusion is reinforced by the findings of Chapter 8, in which it is shown that none of the recently developed fast correlation methods offer significant computational advantages over the well-known FFT method (at least for the quasi-periodic signals encountered in pitch estimation). Thus, those pitch estimation methods which require one or more frequency domain transformations are not suitable when the pitch trajectory spans more than two or three octaves. In addition to these important speed and cost considerations, it should be remembered that some of the successful speech pitch analysis methods (such as cepstrum and inverse filtering analysis) do not work well for more general signals because they rely on signal characteristics which are specific to speech.

This chapter describes two computationally efficient algorithms which operate over wide classes of signals, of

which those encountered in speech and music are typical. The desideratum here was to produce a pitch estimation system which operates successfully on both speech and music signals whose pitches span a six octave range, from 40 Hz to 2.5 kHz. The application initially envisaged was the recording of music by the Piano Typewriter system (which is described in Chapter 3). Thus, musical instruments (including the voice humming or singing) could be interfaced to the system to provide an alternative to the organ keyboard. Such a system must satisfy the following requirements:

(i) It must be inexpensive.

(ii) It should operate rapidly, preferably in real time.

(iii) If incapable of operating in real time, it must provide storage for reasonable signal durations (say, half a minute).

(iv) It must possess a pitch frequency resolution of the order of 1% to 2% throughout the operating pitch range.

(v) It must be able to resolve to within 20 ms to 25 ms temporal events such as the beginning and end of voicing or notes.

Historically, the development of such a system was approached by searching for an established technique which could be suitably modified or extended to satisfy the requirements listed above. The only method which seemed appropriate was Gold's (1962b) method (cf. Section 7.8). Since this method and several simplifying modifications are described in detail by Gold and Rabiner (1969) it is

herein referred to as the Gold and Rabiner (G.R.) algorithm. The G.R. algorithm was suitably modified, implemented in software, and extensively tested using a variety of synthetic and real musical instrument signals. This work is described in Sections 9.3 and 9.6. It was soon obvious that real-time software implementation is not possible, even when special-purpose hardware is incorporated to perform the necessary and time-consuming signal pre-processing. The development of a "stand alone" real-time hardware unit was examined in detail and a preliminary design was made, based on standard TTL logic. However the anticipated hardware cost (approximately $1000 in early 1975) exceeded the budget available to me. Consequently I decided to concentrate on the development of a fast software implementation using the computing facilities already available within the Department. The solution of the problems encountered in attempting to satisfy both of the requirements (iii) and (iv) above is discussed in Section 9.3, wherein is described the storage of a compact representation of the signal on digital magnetic tape for subsequent fast processing.

The decision to implement the G.R. algorithm in software rather than hardware was fortuitous, because it led to the development of a new algorithm which operates successfully on signals which contain more than one prominent peak in each pitch period (see Figure 9.2), and for which the modified G.R. algorithm fails (see Table 9.1). This algorithm is presented in Section 9.4 and its performance is compared with that of the modified G.R.

method in Section 9.6. A further theoretical advantage of the new algorithm is that it is analytically relatable to autocorrelation analysis - the complicated structure of the G.R. algorithm makes such a comparison very difficult for the latter. This analytic relationship between the new algorithm and autocorrelation is discussed in Section 9.5. The relative simplicity of the new algorithm permits its implementation (in hardware or software) to be less complicated than that required for the G.R. algorithm. Specific suggestions for the development of a system which achieves real-time operation are made in Section 9.8.

The application of any pitch estimation method to a music processing system such as the "Piano Typewriter" requires that the pitch trajectory be quantised to the notes of a musical scale. Section 9.7 describes an algorithm which performs this pitch frequency quantisation. This algorithm incorporates a "tune up" feature which permits the successful recognition of notes played by instruments tuned to equally-tempered scales other than the standard A4 at 440 Hz.

It is worth noting that the inclusion of the pitch trajectory quantisation facility and the ability to handle a six octave pitch range is also an advance on previously reported pitch estimators designed specifically for music (Seegar, 1951, 1957; Tove *et al.*, 1966; see also the review by Kassler and Howe, 1975).

9.2 METHODS BASED ON SINGLE PRIMARY FEATURES

Before proceeding to a description of the modified Gold and Rabiner algorithm or the new algorithm, it is worth considering the various signal "features" which can be used to estimate the periodicity of a quasi-periodic signal. This discussion serves to unify the basis of these two algorithms, as well as the other techniques described in Section 7.8.

Consider an arbitrary signal which has been multiplied by a time window whose duration is appropriate to pitch estimation (cf. Section 7.4.1), and biased so that its mean value is zero. Denote this signal by the real continuous time function $s(t)$. A pair of adaptive thresholds $\beta_-(t) < 0$ and $\beta_+(t) > 0$ can be suitably adjusted to partition $s(t)$ into a sequence of contiguous, non-overlapping pulses $p_m(t)$, where $M \in \{1, 2, \ldots M\}$. The pulses are either entirely positive or entirely negative. The duration τ_m of $p_m(t)$ is given by

$$\tau_m = t_{m,2} - t_{m,1} \tag{9.1}$$

where $t_{m,1}$ is the instant at which $s(t)$ either crosses $\beta_+(t)$ with positive slope, or crosses $\beta_-(t)$ with negative slope, and $t_{m,2}$ is the instant at which $s(t)$ next crosses the same threshold. Thus, $s(t)$ can be written as

$$s(t) = \sum_{m=1}^{M} p_m(t) + b(t) \tag{9.2}$$

where the "background" $b(t)$ is the part of $s(t)$ between the thresholds, and does not overlap any of the pulses.

Define the time of occurrence, t_m, of the m^{th} pulse to be the instant at which $|p_m(t)|$ is largest. Other possible choices for t_m are the threshold crossing instants $t_{m,1}$ or $t_{m,2}$ - experience suggests that the peak of $|p_m(t)|$ should be used, because t_m defined in this way exhibits less temporal variation than $t_{m,1}$ or $t_{m,2}$ (cf. Rabiner, Cheng, Rosenberg and McGonegal, 1976). The amplitude (which can be negative or positive) of the m^{th} pulse is defined by

$$A_m = p_m(t_m) \quad . \tag{9.3}$$

The energy in the m^{th} pulse is defined to be

$$E_m = \int_{t_{m,1}}^{t_{m,2}} p_m^2(t)\, dt \quad . \tag{9.4}$$

These definitions are illustrated in Figure 9.1. Since $\{A_m\}$, $\{E_m\}$ and $\{\tau_m\}$ are measurable directly from s(t) they are called the "primary features" of s(t). Define $\underset{\sim}{Y}_m$ to be the vector whose components are A_m, E_m and τ_m. Then the output of the preprocessor which extracts these features from the signal can be represented by the "vector signal" $\underset{\sim}{V}(t)$, where

$$\underset{\sim}{V}(t) = \sum_{m=1}^{M} \underset{\sim}{Y}_m \, \delta(t - t_m) \quad . \tag{9.5}$$

The incorporation of delta functions in equation (9.5) emphasises that the bandwidths of the sampling circuits in the preprocessor must be much larger than the highest significant frequency in s(t) .

It is worth noting that the number of components of $\underset{\sim}{Y}_m$ can be increased, for example by incorporating more thresholds into the preprocessor. However there seems to

be little point in doing this, because the three components A_m, E_m and τ_m contain sufficient information to permit the estimation of the pitch trajectories of most speech and music signals (see Sections 9.4 and 9.6).

If only one primary feature is extracted from each pulse, the vector $\underset{\sim}{Y}_m$ (equation 9.5) can be replaced by the scalar y_m which corresponds to the chosen primary feature. The output of the preprocessor is conveniently denoted by $v(t)$, because it is now a scalar.

To determine the pitch periods of $s(t)$ it is required that the pulses be grouped in recurring pairs. A simple way of achieving this is to evaluate

$$g_{m,\nu} = \left| (y_m - y_{m-\nu}) / (y_m + y_{m-\nu}) \right| \qquad (9.6)$$

for $\nu = 1, 2, \ldots (m-1)$, and to postulate that $p_m(t)$ is a recurrence of $p_{m-\nu}(t)$ if $g_{m,\nu} < \varepsilon$, where ε is a positive real number which is chosen to accord with the expected variations between successive recurring pulses. An estimate at time t_m of the period T of $s(t)$ is given by

$$T_m = t_m - t_{m-\nu_m} \qquad (9.7)$$

where ν_m is the smallest positive integer value of ν for which $g_{m,\nu} < \varepsilon$. The estimated pitch trajectory samples T_m obtained in this way can be subsequently smoothed (see Section 9.7; also Rabiner, Sambur and Schmidt, 1975).

Two conditions must be satisfied for a single-feature algorithm based on the relations (9.6) and (9.7) to be useful:

(a) When s(t) is truly periodic, a single feature is sufficient to distinguish between the individual pulses which constitute one period of s(t).

(b) The variations between the waveforms of adjacent periods are sufficiently small that ε is not required to be so large that erroneous "pulse pairings" occur too frequently.

It is worth noting here that if the signal occupies only a small percentage bandwidth (e.g. those used in most carrier communication systems) then the signal consists of only two pulses in each period and the zero crossing rate is twice the frequency. This observation remains essentially valid even in the presence of considerable noise. For such signals both of the conditions (a) and (b) are satisfied. The most suitable y_m is τ_m, in which case the algorithm is equivalent to counting zero crossings.

It is also worth commenting that many early pitch estimation methods attempt to preprocess s(t) so that only two zero crossings occur in each period. Such preprocessing techniques are considered in detail by McKinney (1965) (see also the second paragraph of Section 7.1).

Few wide band signals of the kind encountered in speech and music satisfy both of the conditions (a) and (b), whichever primary feature is represented by y_m. For example, if $y_m = A_m$ then condition (a) is not satisfied if two or more pulses within a single period have the same amplitude. Even when these amplitudes are appreciably different, the envelope variations that occur in practice force ε to be so large that condition (b) is often violated.

This is illustrated in Figure 9.2.

The single feature algorithm outlined in relations (9.6) and (9.7) can be usefully extended to operate successfully over a wide class of signals of practical interest by making a number (L, say) of comparisons of each y_m with a number (K, say) of its predecessors. The output of the preprocessor can then be represented by the vector signal $\underset{\sim}{W}(t)$, where

$$\underset{\sim}{W}(t) = \sum_{m=1}^{M} \underset{\sim}{Z}_m \, \delta(t - t_m) \tag{9.8}$$

where the dimension of $\underset{\sim}{Z}_m$ is KL. The $(k-1+\ell)^{th}$ component of $\underset{\sim}{Z}_m$ is a particular comparison of y_m with y_{m-k}. For example, for $\ell = 1$ the comparison might be whether or not $|y_m|$ is greater or less than $|y_{m-k}|$, for $\ell = 2$ the comparison might be of $\text{sgn}(y_m)$ and $\text{sgn}(y_{m-k})$, etc. The parallel processing algorithms of Gold (1962b) and Gold and Rabiner (1969) are of this kind (see also Sections 7.8 and 9.3). These algorithms use $y_m = A_m$. The observation that the G.R. method fails for signals of the kind depicted in Figure 9.2 prompted the development of an algorithm which uses all three of the signal primary features defined in equations (9.1), (9.3) and (9.4). This algorithm is described in Section 9.4.

9.3 THE MODIFIED GOLD AND RABINER ALGORITHM

The original Gold and Rabiner (1969) parallel processing algorithm was designed primarily for speech, and operates in four stages.

1. The speech signal is filtered to select approximately the first formant region. Typically, a band pass filter with cut-off frequencies at 100 Hz and 600 Hz is used.
2. The amplitudes and times of occurrence of the maxima and minima of the filtered signals are measured. From these, six impulsive signals are generated.
3. Six identical pitch period estimators, each operating on one of the six impulsive signals, produce independent period estimates.
4. The independent period estimates are used to produce a final pitch period estimate by employing a relatively sophisticated "voting" algorithm.

The original G.R. algorithm was modified to permit analysis over a six octave pitch range. This modified algorithm is now described, and its implementation discussed in terms of the requirements laid down in Section 9.1.

For speech or music signals whose pitches span five or six octaves, the filtering employed in the G.R. method is inappropriate. However, it is sometimes advantageous to eliminate hum and low frequency noise by employing a high-pass filter which excludes frequencies from 0 Hz to about 80 Hz or 100 Hz. The spectral magnitude response of this filter is not critical, although it is desirable that the phase response be approximately linear, as Miller (1975) has pointed out.

The magnitude A_m and time of occurrence t_m of each pulse $p_m(t)$ of the wideband signal s(t) is measured. The terminology used here is the same as that introduced in Section 9.2. In this case the positive and negative

thresholds $\beta_+(t)$ and $\beta_-(t)$ are set to constant values which just exceed the average background noise level.

For each positive pulse $p_m(t) > 0$ three scalar quantities $F_{i,p}$ are generated. Similarly, for each negative pulse $p_m(t) < 0$ three scalar $F_{i,n}$ are generated. These $F_{i,p}$ and $F_{i,n}$ correspond to peak and peak-to-peak measurements, and are generated as follows. Denote by A_p the most recent A_m such that $A_m > 0$, and by A_n the latest $A_m < 0$. Distinguish the t_m which correspond to positive and negative pulses by defining instants t_p and t_n such that

$$A_p = p_m(t_p)$$

and

$$A_n = p_m(t_n) \ . \tag{9.9}$$

Then

$$\begin{aligned} F_{1,p} &= A_p \\ F_{2,p} &= A_p + |A_n| \\ F_{3,p} &= A_p - A_{p-1} \quad \text{if} \quad A_p > A_{p-1} \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{9.10}$$

These features "occur" at time t_p. Similarly

$$\begin{aligned} F_{4,n} &= |A_n| \\ F_{5,n} &= |A_n| + A_p \\ F_{6,n} &= |A_n| - |A_{n-1}| \quad \text{if} \quad |A_n| > |A_{n-1}| \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{9.11}$$

The $F_{i,n}$ "occur" at time t_n.

Six scalar functions $v_i(t)$ are formed from the $F_{i,p}$ and F_{in}. Thus

$$v_i(t) = \sum_{p=1}^{P} F_{i,p}\, \delta(t - t_p) \qquad i = 1,2,3$$

and

$$v_i(t) = \sum_{n=1}^{N} F_{i,n}\, \delta(t - t_n) \qquad i = 4,5,6 \quad . \tag{9.12}$$

The choice of this set of measurements is based on consideration of two extreme cases illustrated in Figure 9.3. For the case when the fundamental only is present, period estimates derived from v_1, v_2, v_4 and v_5 are correct but those from v_3 and v_6 fail (see Figure 9.3a). When a strong second harmonic exists with weak fundamental, period estimates from v_3 and v_6 are correct whereas those from v_1, v_2, v_4 and v_5 incorrectly indicate half the period (see Figure 9.3b). Despite this, the final period computation (described below) has a high probability of being correct in both cases.

Independent estimates $T_{i,k}$ of the periodicity of each $v_i(t)$ are computed using a "blanking interval" followed by a decaying threshold. The threshold time constant depends on the smoothed $\bar{T}_{i,k}$ which is computed from previous $T_{i,k}$. Thus for each $v_i(t)$

$$T_{i,k} = t_{(p_n),k} - t_{(p_n),k-1} \tag{9.13}$$

where $t_{(p_n),k}$ is the instant at which the k^{th} match is detected from the threshold crossing

$$v_i(t) > \theta_i(t) \quad . \tag{9.14}$$

The threshold $\theta_i(t)$ following the k^{th} match is defined by

$$\theta_i(t) \rightarrow \infty \quad \text{for} \quad t_{(p_n),k} \leq t < t_{(p_n),k} + \tau_i$$

$$\theta_i(t) = v_i(t_{(p_n),k}) \exp[-\{t - (t_{(p_n),k} + \tau_i)\} / \beta_i]$$

$$\text{for} \quad t_{(p_n),k} + \tau_i \leq t < t_{(p_n),k+1}$$

$$i = 1, 2, \ldots 6 \tag{9.15}$$

and is reset each time a match is detected. The blanking interval duration τ and time constant β are given by

$$\tau_i = 0.375 \, \bar{T}_{i,k}$$

$$\beta_i = \bar{T}_{i,k} / 0.695 \tag{9.16}$$

where $\bar{T}_{i,k}$ is computed from

$$\bar{T}_{i,k} = (\bar{T}_{i,k-1} + T_{i,k}) / 2 \tag{9.17}$$

when the k^{th} match is detected. To prevent extreme values of τ and β, $\bar{T}_{i,k}$ is constrained to the range 0.4 ms to 25.5 ms, which corresponds the pitch frequency range 40 Hz to 2.5 kHz. The definition of τ used here is slightly less than the $0.4 \, \bar{T}_{i,k}$ used in the original G.R. algorithm, because this improves the detector performance when the signal possesses a strong third harmonic component.

Observe that the process used to estimate $T_{i,k}$ from $v_i(t)$ can be viewed as a simple cross correlation of the $F_{i,p}$ or $F_{i,n}$.

Once every 5 ms the $T_{i,k}$ are combined into a "vote matrix" M which is formed as follows:

$$
\begin{aligned}
M_{1,i} &= T_{i,k} \\
M_{2,i} &= T_{i,k-1} \\
M_{3,i} &= T_{i,k-2} \\
M_{4,i} &= T_{i,k} + T_{i,k-1} \\
M_{5,i} &= T_{i,k-1} + T_{i,k-2} \\
M_{6,i} &= T_{i,k} + T_{i,k-1} + T_{i,k-2}
\end{aligned}
\qquad (9.18)
$$

where $T_{i,k}$ is the most recent period estimate from the i^{th} detector. Note that if entries in the first three rows incorrectly measure the second harmonic, then the corresponding entries in rows four and five are correct. Similarly, entries in row six are correct if the corresponding entries in rows one to three incorrectly estimate the third harmonic. In addition, the non-linearity in the computation of features of type 3 and 6 (cf. equations 9.10 and 9.11) ensures that $T_{3,k}$ and $T_{6,k}$ are likely to be correct if significant second or third harmonic exists in s(t) (see Figure 9.3).

To compute the final pitch period estimate from the vote matrix, each $M_{1,i}$ in turn is compared with all other $M_{j,k}$. The total number of "coincidences" (defined below)

is counted. The entry $M_{1,i}$ (i = 1, 2, ... 6) which registers the greatest number of coincidences is the final period estimate.

If a tie is encountered where several candidates "win", a secondary vote procedure is used. "Winning" candidates are sorted into groups of mutually-coincident "winners", and the mean value of that group with the largest number of members is the final period estimate. If no group "wins", a "doubtful" vote is registered and the mean value of the group nearest to the previous period estimate is used as the final estimate. If a "doubtful" vote occurs for two successive period calculations, a "noise" decision is made (analogous to the "unvoiced" decision in speech), whereupon all detectors are reset and the vote matrix memory is cleared. The detector reset operation (which is also performed during the pitch estimator initialisation, and when a "silence" condition is detected) sets $\bar{T}_{i,k}$ = 0.4 ms so that τ and β (equation 9.16) are minimised. In addition, $\theta_i(t)$ is set to zero. Experiments with signals whose spectral compositions span a wide range show that when the reset operation is performed during analysis of a steady state signal, detector equilibrium is usually re-established within two pitch periods.

The period estimate coincidence measure used should ideally be based on the ratio of the two estimates being compared. Thus, if $M_{1,i}$ and $M_{j,k}$ are being compared, a coincidence could be defined by

$$\left| \frac{M_{j,k} - M_{1,i}}{M_{j,k}} \right| < \varepsilon_1 \qquad (9.19)$$

or alternatively by

$$\left| \frac{M_{j,k} - M_{1,i}}{M_{j,k} + M_{1,i}} \right| < \varepsilon_2 . \qquad (9.20)$$

Since many (6 × 35) coincidence calculations are required to produce each final period estimate, a simpler coincidence criterion is used, to avoid the need for a divide operation:

$$\left| M_{j,k} - M_{1,i} \right| < \varepsilon_3(M_{1,i}) . \qquad (9.21)$$

A table look-up procedure is used to obtain $\varepsilon_3(M_{1,i})$ - the values used are listed in Table 9.3. Note that ε_3 need be calculated only six times during the entire pitch period calculation, since each $M_{1,i}$ is compared with 35 different $M_{j,k}$.

It is in the final pitch period calculation that the greatest departure from the original G.R. algorithm occurs. The original G.R. algorithm does not use the secondary vote procedure to eliminate ties, but instead performs the initial vote using four different values of $\varepsilon_3(M_{1,i})$. For each vote a bias is subtracted, so that fine thresholds are weighted more heavily than coarse thresholds. This extended vote procedure requires 6 × 4 × 35 coincidence calculations for each final period calculation, and is designed to accommodate the rapid waveshape transitions encountered with speech. Gold and Rabiner report that their software implementation of the original algorithm runs about 50 times slower than real time.

Gold and Rabiner also propose two modifications of

the original algorithm, which they report to be suitable for speech whose highest pitch is limited to 220 Hz and 300 Hz respectively. These modifications use a single value of ε_3 which is independent of $M_{1,i}$ (cf. equation 9.21) and fixed values of τ and β (equation 9.16). The second modification also reduces the number of estimators from six to four, and simplifies the voting matrix by omitting rows 3, 5 and 6. While these modifications result in software implementations which run typically 1.3 to 1.5 times slower than real time, they are unsuited for music because they are designed specifically for a narrow pitch frequency range.

Three different software implementations of the modified G.R. algorithm described above have been made. The first implementation was an attempt to achieve real-time operation. Special hardware peak detectors are patched on the EAI 580 analogue computer, to measure A_p and A_n. These quantities are read by the 640 digital machine (via an ADC) whenever a peak detector interrupt is generated. A peak detector interrupt also controls a pair of special hardware clocks, so that the intervals between successive t_p and t_n are measured. These intervals are read by the 640 via the RTL interface (cf. Section 3.3). A fast ASSEMBLY language program was written, using fixed-point arithmetic, to simulate the six individual estimators and to perform the final period calculation. To reduce the time required to evaluate the threshold $\theta_i(t)$ (cf. equation 9.15) a fast formula based on a piecewise linear approximation to a normalised exponential function is used - this is described below. The voting matrix is generated and the final period

estimate is computed once every 10 ms. This process is initiated by an interval timer interrupt.

The implementation described above does not achieve real-time operation over a useful pitch range. The peak interrupt servicing routine and the associated detector processing requires about 800 μs execution time, despite the use of very efficient machine-level programming. Thus the minimum allowable interval between two successive positive peaks is 1600 μs, which corresponds to a maximum permitted fundamental frequency of 625 Hz for a pure sinusoid. This is reduced by factors of 2 and 3 respectively if significant second or third harmonic is present. Since this calculation does not include the final period calculation, it is obvious that real-time operation is not practicable using this approach. This conclusion should however be qualified by the observation that the most critical bottleneck in the program is a loop which contains several "multiply" operations (each of which requires approximately 18 μs). The use of a faster computer (such as the GTE Sylvania programmable signal processor, which requires only 750 ns for each multiplication - cf. Ross *et al.*, 1974) would permit the existing implementation to achieve real-time operation for a useful range of pitch frequencies.

A major problem encountered with slower than real-time operation lies in the necessity to store the signal $s(t)$ or a suitably preprocessed representation of it (e.g. the $\underset{\sim}{V}(t)$ defined in equation 9.5). The storage of digital samples of $s(t)$ requires a large memory if useful signal

durations are accommodated - the interdependence of sampling rate, pitch frequency resolution, and pitch range which is discussed in Section 7.4.2 (in the paragraphs containing equations 7.12 to 7.14) apply here. The use of analogue magnetic tape storage is inconvenient because of the need for synchronisation and control to permit the computer to sample and process successive sections of signal. The digital magnetic tape storage facility available incorporates the necessary control facilities and provides storage for 150K words, but is limited to a word write rate of about 5.0 kHz (Jordan, 1974). By using only 8 bits for each sample an effective sampling rate of about 10.0 kHz is achieved. However the restriction to 8 bits places a severe limitation on the signal dynamic range which can be accommodated without excessive quantisation noise (cf. Section 6.3). Another storage medium available is the fixed head disc which provides storage for a total of 360K words. Unfortunately a DMA facility is not incorporated, so that signal samples cannot be stored on disc without interrupting the signal sampling process for an unacceptably long period. Thus the largest available memory which is suitable as buffer storage of uniformly-spaced signal samples acquired at useful rates (20 kHz to 40 kHz) is the 16K words core storage.

The existence of the signal sampling, editing and disc storage system which is described in the Appendix led to the development of the second implementation of the modified G.R. algorithm. This signal acquisition system permits the storage on disc of 12K uniformly-spaced samples,

using a core buffer store. The sampling rate used is controllable, up to a maximum rate of about 40K samples per second. The second implementation of the modified G.R. algorithm is a FORTRAN IV program which reads successive "blocks" of 880 samples from disc to core and measures digitally the magnitude and time of occurrence of each A_n and A_p. It then simulates the six independent period estimators, performs the final period calculation, and plots the resulting pitch trajectory on the storage oscilloscope. This version was used to evaluate the modified G.R. algorithm and to produce the results presented in Section 9.6. It requires approximately 25 seconds to process 1 second of signal, although the actual processing time depends upon the rate at which peaks occur. The main limitation of this implementation is the restricted signal duration (less than 2 s, depending on the sampling rate used).

The third implementation uses the digital magnetic tape to store a coded representation of the scalar signal

$$v(t) = \sum_{m=1}^{M} A_m \, \delta(t - t_m) \qquad (9.22)$$

where A_m and t_m are defined in Section 9.2. The magnitude (which may be positive or negative) and time of occurrence of each A_m is stored as the pair of numbers (DTPP, PKMAG). DTPP is the number of samples corresponding to the interval $(t_m - t_{m-1})$ while PKMAG is the peak magnitude, A_m. PKMAG is constrained to be non-zero, unless a peak has not been detected within the last 20 ms. When this latter condition occurs the entry (DTPPMX, 0) is stored, where DTPPMX

corresponds to an interval of 20 ms. This prevents DTPP from becoming sufficiently large that an overflow condition occurs. The present system which creates the digital magnetic tape signal representation operates by sampling the signal at uniform intervals. It then computes the entries (DTPP, PKMAG). The latter are stored in core in a large circular buffer, the entries of which are written at uniform intervals on to the magnetic tape. Since this system is implemented entirely in software, the highest signal sampling rate which is possible is about 8.3 kHz (which is too low to achieve adequate pitch frequency resolution). A hardware digital peak detector has been designed to permit (DTPP, PKMAG) to be measured using a signal sampling rate of 100 kHz. This hardware generates an interrupt each time a peak is detected. In this way, the computer need only read DTPP and PKMAG and perform the housekeeping operations required to ring-buffer the output to the magnetic tape.

The pitch trajectory is computed and displayed using software which is essentially a fast version of that used in the second implementation. The main difference is that the data are read from tape in blocks of 3000 words, the peak detector is omitted, and the threshold $\theta_i(t)$ (equation 9.15) is computed using a fast piecewise-linear approximation to the normalised exponential, exp[-t/0.695]. This approximation uses the following break-point values:

t	exp[-t / 0.695]	
0 * $\bar{T}_{i,k}$	1.0	
1 * $\bar{T}_{i,k}$	0.5	
2 * $\bar{T}_{i,k}$	0.25	
3 * $\bar{T}_{i,k}$	0.125	etc.

where * denotes multiplication and the time origin is set at $t_{(p_n),k} + \tau_i$. This approximation to the normalised exponential possesses a worst case error of about 5%.

An additional difference from the second implementation is that efficient in-line ASSEMBLY language coding and fixed-point arithmetic are used extensively to improve speed.

The resulting pitch trajectory is either displayed or is quantised to the equally-tempered scale using the algorithm described in Section 9.7. This latter facility permits a table of note events to be constructed in the MOD data format (cf. Section 3.4). This table is subsequently stored on disc in the same form used by the Piano Typewriter files. Figure 9.6 illustrates these features, and their application to the Piano Typewriter system - Figure 9.6(d) is the edited TRAD notation copy produced from the pitch trajectory shown in Figure 9.6(a). In this case the input sound is a female humming, and a condenser microphone is used. Useful results have also been obtained from both violin and clarinet sounds, using either a condenser microphone or special lightweight motion-sensitive transducers (e.g. those marketed by Barcus Berry) which

mount on the bridge or reed, respectively. These latter transducers are especially useful, because they permit the sounds produced by each individual instrument in an ensemble to be monitored with a high signal-to-noise ratio not achievable with microphones.

9.4 THE SECONDARY FEATURE ALGORITHM

The modified G.R. algorithm described in the preceding section works well for many speech and music signals. However it fails for signals which possess several peaks of similar magnitude in one pitch period (see Figure 9.2 and Table 9.1). This section describes a new algorithm which uses all three of the primary features defined in Section 9.2 and which successfully operates on such waveforms.

The simple algorithm which is described in the paragraph containing equations (9.6) and (9.7) can be extended to use all three primary features. However, the improvement is not great, mainly because the A_m are strongly affected by envelope variations. This suggests that "secondary features" which are relatively insensitive to envelope variations should be derived from the primary features. Something which is completely insensitive to envelope variation is

$$B_m = \mathrm{sgn}(A_m) \quad . \tag{9.23}$$

Another quantity little affected by the signal envelope is

$$C_m = A_m \; / \; E_m^{\frac{1}{2}} \quad . \tag{9.24}$$

Of the primary features, τ_m is the least sensitive to envelope variations. While B_m and C_m are of the same sign and could therefore be combined into a single secondary feature, it is advantageous to distinguish them. Thus, B_m can be incorporated into the processor memory address structure so that positive and negative pulses are stored and processed separately.

The quantities B_m, $|C_m|$ and τ_m are satisfactory secondary features. It is also useful to include E_m, even though this is sensitive to envelope variations, because it can provide a useful final check on whether a particular pulse is actually a recurrence of a previous pulse. E_m is also used in the computation of the final pitch period estimate.

The preprocessor output vector signal $\underset{\sim}{V}(t)$ which is defined in equation (9.5) is replaced by another vector signal

$$\underset{\sim}{U}(t) = \sum_{m=1}^{M} \underset{\sim}{X}_m \, \delta(t - t_m) \tag{9.25}$$

where the components $X_{\ell,m}$ of the vector $\underset{\sim}{X}_m$ are the secondary features:

$$X_{1,m} = B_m \, , \quad X_{2,m} = |C_m| \, , \quad X_{3,m} = \tau_m \, , \quad X_{4,m} = E_m \, . \tag{9.26}$$

For $\ell > 1$ the quantities

$$G_{\ell,m,\nu} = |(X_{\ell,m} - X_{\ell,m-\nu}) / (X_{\ell,m} + X_{\ell,m-\nu})| \tag{9.27}$$

are computed. Small positive real numbers $\varepsilon_{\ell,\nu}$ are chosen so that $p_m(t)$ can be said to be a recurrence of $p_{m-\nu}(t)$ if

$$X_{1,m} = X_{1,m-\nu}$$

and

$$G_{\ell,m,\nu} < \varepsilon_{\ell,\nu} \quad \text{for} \quad \ell = 2,\ 3 \text{ and } 4 \ .$$

Ideally, the $\varepsilon_{\ell,\nu}$ should be made small for values of ν less than the expected number of pulses per period. This applies particularly to $\varepsilon_{4,\nu}$ because E_m can be expected to vary appreciably from one period to the next. An *a priori* estimate of the expected pitch period of the current signal segment could be obtained by extrapolating the smoothed pitch trajectory (see also Section 7.4.6). However, this refinement has not been included in the present algorithm, which uses

$$\varepsilon_{2,\nu} = \varepsilon_{3,\nu} = 0.1 \quad \text{and} \quad \varepsilon_{4,\nu} = 0.3 \ .$$

An estimate of T at time t_m is given by T_m as defined by equation (9.7), where ν_m is the least positive integer value of ν for which $X_{1,m} = X_{1,m-\nu}$ and $G_{\ell,m,\nu} < \varepsilon_{\ell,\nu}$ for all three values of $\ell > 1$.

The pitch trajectory can be constructed by suitably smoothing the pitch estimates T_m which occur at t_m. However, when the pitch of $s(t)$ is approximately constant a better estimate of T is given by

$$\tilde{T} = \left(\sum_{m=1}^{M} (E_m E_{m-\nu_m})^{\frac{1}{2}} T_m \right) \Big/ \sum_{m=1}^{M} (E_m E_{m-\nu_m})^{\frac{1}{2}} \ . \qquad (9.28)$$

The justification for this is given in Section 9.5. Since the pitch is presumed to be approximately constant throughout the signal segment which contains the M pulses,

those estimates which are "excessively erroneous" should be identifiable. This identification is achieved by calculating the real, positive number $\hat{T}$ which is closest to most of the T_m. It is worth noting that $\hat{T}$ is the average of the T_m only when all of the latter are equal, and that a simple method of computing $\hat{T}$ is to construct a discrete period histogram and locate its mode. $\hat{T}$ is then used to remove from the summations in equation (9.28) those terms which correspond to values of m for which

$$|T_m - \hat{T}| > T\Delta \quad , \tag{9.29}$$

where Δ is a specified fractional tolerance. A suitable value for Δ is 0.05.

9.5 COMPARISON OF SECONDARY FEATURE ALGORITHM WITH AUTOCORRELATION ANALYSIS

Denote by $\psi(\tau)$ and $\psi_{m,n}(\tau)$ the autocorrelation of $s(t)$ and the cross-correlation of the m^{th} and n^{th} pulses respectively. The autocorrelation (denoted by $\phi(\tau)$) of $[s(t) - b(t)]$ can be written in the form

$$\phi(\tau) = \sum_{m,n=1}^{M} \psi_{m,n}(\tau) \quad . \tag{9.30}$$

The correlation of two functions of finite duration, or of two finite sequences, exists for no longer than the sum of the durations of the functions, or the sum of the extents of the sequences. Consequently the limits appended to any integrals or summations representing such correlations must depend explicitly upon the correlation variable (here the

delay τ). Any precise notation is therefore ungainly, and the notation of equation (9.30) is used here.

When s(t) is truly periodic, $\phi(\tau)$ is identical with

$$\hat{\phi}(\tau) = \sum_{m=1}^{M} \psi_{m,m-\bar{\nu}}(\tau) \qquad (9.31)$$

within the delay interval which is centred at $\tau = T$ and which is of duration one half of the shortest of the τ_m. Note that $\bar{\nu}$ is the least positive integer which satisfies

$$G_{\ell,m,\bar{\nu}} = 0 , \quad m \in \{1,2, \ldots M\} , \quad \ell \in \{1,2,3,4\}. \qquad (9.32)$$

Recall that $\phi(\tau)$ has an absolute maximum at $\tau = 0$, and that the period of s(t) cannot be less than the shortest interval (denoted by τ_o) between the start of a positive pulse and the end of the next negative pulse. If s(t) is truly periodic and free of noise then the period of s(t) is that value of $\tau > \tau_o$ at which $\phi(\tau)$ has its largest positive value. In practice, however, it is possible for $\phi(\tau)$ to be larger in the neighbourhood of $\tau = \ell T$ where ℓ is an integer greater than unity, than it is in the neighbourhood of $\tau = T$, where T is the true period. This applies particularly when the duration of s(t) spans many periods, so that the linear taper on $\phi(\tau)$ is not steep. Thus T can be estimated by observing the separation of the major positive peaks of $\phi(\tau)$. Alternatively, a positive threshold $\hat{\beta} < \phi(0)$ can be introduced, where $(1 - \hat{\beta} / \phi(0))$ is a little larger than the estimated signal-to-noise ratio. The period T is estimated from

$$\phi(T) = \max \phi(\tau) , \quad \tau_- < \tau < \tau_+ \qquad (9.33)$$

where τ_- is the least value of $\tau > \tau_o$ at which $\phi(\tau)$ crosses $\hat{\beta}$ with positive slope, and τ_+ is the least value of $\tau > \tau_-$ at which $\phi(\tau)$ crosses $\hat{\beta}$ with negative slope.

When s(t) is a section of a real world signal it is possible to write

$$\phi(\tau) = \hat{\phi}(\tau) + \theta(\tau) \quad , \quad \tau_- < \tau < \tau_+ \tag{9.34}$$

where $\hat{\phi}(\tau)$ is defined in equation (9.31), and where $\theta(\tau)$ has the character of noise because it includes some $\psi_{m,n}(\tau)$ not included in equation (9.31). The integer $\bar{\nu}$ appearing in (9.31) must now be taken as the average of the ν_m appearing in equation (9.28).

It is convenient to define real numbers η_m such that

$$\psi_{m,m-\bar{\nu}}(T) = (1 + \eta_m / T) \; (E_m E_{m-\bar{\nu}})^{\frac{1}{2}} \tag{9.35}$$

where T, the estimate of the period as given by auto-correlation analysis, is defined by equation (9.33). When s(t) is truly periodic, $E_{m-\bar{\nu}} \doteq E_m$ so that the definitions of $\psi_{m,n}(\tau)$ and E_m ensure that $\eta_m = 0$. Thus for real-world signals the η_m are measures of the differences between recurring pulses in successive periods of s(t). On substituting equations (9.31) and (9.35) into (9.34) and multiplying through by T we get

$$T \; \phi(T) = \sum_{m=1}^{M} (E_m E_{m-\bar{\nu}})^{\frac{1}{2}} \; \hat{T}_m + T \; \theta(T) \tag{9.36}$$

where

$$\hat{T}_m = T + \eta_m \quad . \tag{9.37}$$

It follows necessarily from the definitions of $\psi_{m,n}(\tau)$, $\phi(\tau)$ and $\hat{\phi}(\tau)$ that

$$\hat{\phi}(0) = \phi(0) = \sum_{m=1}^{M} E_m \quad . \tag{9.38}$$

When $\tau = T$, most of the $\psi_{m,m-\bar{\nu}}(\tau)$ are close to their largest values. Consequently, if $\phi(T)$ is written in the form

$$\phi(T) = \gamma \sum_{m=1}^{M} (E_m E_{m-\bar{\nu}})^{\frac{1}{2}} \tag{9.39}$$

then the positive real number γ is expected to be less than unity by an amount equal to the effective signal-to-noise ratio. Combining equations (9.36) and (9.39) gives

$$T = \left(\sum_{m=1}^{M} (E_m E_{m-\bar{\nu}})^{\frac{1}{2}} \; (\hat{T}_m / \gamma) \right) / \sum_{m=1}^{M} (E_m E_{m-\bar{\nu}})^{\frac{1}{2}} + \Phi \tag{9.40}$$

where the "noise" Φ is given by

$$\Phi = T\,\theta(T) / \gamma \sum_{m=1}^{M} (E_m E_{m-\bar{\nu}})^{\frac{1}{2}} \quad . \tag{9.41}$$

Multiply the numerator and denominator of the first term on the right-hand side of equation (9.40) by the positive real number α, where

$$\alpha \sum_{m=1}^{M} (E_m E_{m-\bar{\nu}})^{\frac{1}{2}} = \sum_{m=1}^{M} (E_m E_{m-\nu_m})^{\frac{1}{2}} \tag{9.42}$$

where the ν_m are defined in the penultimate paragraph of Section 9.4. There must exist positive real numbers, λ_m say, such that

$$\Phi = \left(\sum_{m=1}^{M} (E_m E_{m-\nu_m})^{\frac{1}{2}} \lambda_m \right) / \sum_{m=1}^{M} (E_m E_{m-\nu_m})^{\frac{1}{2}} \quad . \tag{9.43}$$

Substituting equations (9.42) and (9.43) into (9.40) gives

$$T = \left(\sum_{m=1}^{M} (E_m E_{m-\nu_m})^{\frac{1}{2}} \tilde{T}_m \right) / \sum_{m=1}^{M} (E_m E_{m-\nu_m})^{\frac{1}{2}} \qquad (9.44)$$

where

$$\tilde{T}_m = (E_{m-\bar{\nu}} / E_{m-\nu_m})^{\frac{1}{2}} (\alpha \hat{T}_m / \gamma) + \lambda_m \quad . \qquad (9.45)$$

The justification for equation (9.28) is now apparent. Its form is similar to that of equation (9.44), which is provided by the "rigorous" method of autocorrelation analysis. The difference between the two estimates of the period T of s(t) is accounted for by the differences between the T_m and the $\tilde{T}_m$. All of these differences vanish when s(t) is truly periodic, in which case $\hat{T}_m = T$ from equation (9.37) because all the η_m are zero, by definition. So, $\tilde{T}_m = T$ from (9.45) because $\alpha = \gamma = 1$ and, for every m, $\lambda_m = 0$ and $\nu_m = \bar{\nu}$; but $T_m = T$ because s(t) is truly periodic.

An error in the pitch period estimate provided by autocorrelation analysis can only be caused by differences in the shapes of recurring pulses. These differences are characterized by the η_m. When the latter are zero then equations (9.35) to (9.37) show that the pitch period estimate is exact.

When the secondary feature algorithm correctly identifies all recurring pulses then $\nu_m = \bar{\nu}$ for every m, so that any differences between the T_m and T is due to the differences in the shapes of recurring pulses. Thus the secondary feature algorithm is virtually equivalent

to autocorrelation analysis in this case. The major sources of discrepancy between $\tilde{T}$ and T are those integers ν_m which do not equal $\bar{\nu}$.

9.6 COMPARATIVE RESULTS

The secondary feature algorithm has been implemented in software on the EAI 640. FORTRAN IV is used, because the main objective was to evaluate and refine the algorithm rather than to achieve fast operation. For this reason also, software floating-point arithmetic is used even though this is at least an order of magnitude slower than fixed-point arithmetic. The signal data acquisition and disc storage system outlined in Section 9.3 is used to provide the input signal samples. Thus, successive blocks of 880 samples are read from disc on to core for subsequent processing.

The implementation is designed to handle signals whose pitches span a six octave range from 40 Hz to 2.5 kHz. Two separate memory stacks are incorporated, so that the features of positive and negative pulses are stored separately. In this way the secondary feature $B_m = X_{1,m}$ is conveniently handled. Since the lowest pitch frequency of interest is 40 Hz, the length of each memory stack is adjusted to discard the features of pulses which occurred earlier than 25 ms before the present instant. Thus the effective signal window is rectangular and has a fixed duration of 25 ms. This interval is also used for the computation of the final period estimate using equation (9.28). It is worth pointing out that the memory

length of the "vote matrix" used in the G.R. algorithm is usually about 3 pitch periods, which corresponds to 25 ms at a pitch frequency of 120 Hz. Therefore the comparative results presented here are fair in the sense that the pitch trajectory smoothing implied by equation (9.28) is applied over an interval which is of similar duration to the G.R. vote matrix memory length. In the present implementation the "nominally correct" period estimate $\hat{T}$ is computed by comparing each T_m with every other T_m using a similar "coincidence" criterion to that used in the G.R. algorithm. In this case the coincidence measure given by equation (9.19) is used, with $\varepsilon_1 = 0.05$. $\hat{T}$ is set equal to that T_m which registers the largest number of coincidences. A simpler and faster method is to construct a discrete period histogram using the T_m, and to set $\hat{T}$ equal to the mode of the histogram.

In the present implementation no distinction is made between the "silence" and "unvoiced" decisions - both are treated as a silence. A silence is registered if no A_m exceeds the threshold β_+ or β_- during the current 25 ms interval (i.e. s(t) consists entirely of "background", b(t) - cf. equation 9.2) or if the largest E_m in any 25 ms interval is less than 0.05 $E_{m_{max}}$, where $E_{m_{max}}$ denotes the largest value of E_m in the previous 25 ms interval.

Results obtained using the secondary feature algorithm are compared with those obtained using the modified G.R. algorithm in Table 9.1 and in Figures 9.4 and 9.5. In these cases the acoustic signals were sensed using a Brüel and Kjaer Type 4134 condenser microphone

whose bandwidth extends from 20 Hz to 20 kHz. The signals were sampled at rates of 42 kHz for musical instruments and 10 kHz for speech, and digitised to 14 bits.

Table 9.1 presents the results obtained from an analysis of several notes played by a selection of orchestral instruments. A 75 ms portion of a steady note is analysed in each case. To obtain a useful number of pitch estimates, the period estimate $\tilde{T}$ is computed at 5 ms intervals. To ensure that the comparison with the G.R. algorithm is fair, the final pitch period averaging (equation 9.28) is performed over a 10 ms interval rather than the usual 25 ms interval. Gross errors in $\tilde{T}$ are defined as those estimates which differ by more than 10% from the eye-detected period. Gross errors are not included in the mean and standard deviation calculations. It is apparent that the secondary feature algorithm produces significantly fewer gross errors than the G.R. algorithm, and that for most notes the pitch frequency estimates exhibit less variation. It is worth remarking that the two trumpet notes which cause the G.R. algorithm to fail (notes 2 and 3) possess waveshapes of the kind illustrated in Figure 9.2(b), for which the G.R. algorithm is expected to fail.

Figures 9.4 and 9.5 illustrate pitch trajectories obtained from utterances by a male speaker. Compare first the secondary feature and G.R. algorithms under the special conditions for which the latter is designed. Before digitisation, the analogue speech signals were prefiltered to the pass band 100 Hz to 600 Hz, which corresponds

approximately to the first formant region. The speech was recorded in a room with high ambient noise level, due mostly to the fans of an air-conditioning system. The average signal-to-noise ratio (before filtering) was 20 dB. Both algorithms performed similarly, as Figure 9.4 shows. The pitches estimated manually from the displayed signal agree with the crosses and circles where they are closely grouped in Figure 9.4. The few erroneous pitch estimates produced by both algorithms are probably of little significance in Vocoder applications, because they occur at the beginning and end of voicing where the signal energy is low. In addition, subsequent pitch trajectory smoothing (for example using the non-linear median smoothing algorithm described by Rabiner, Sambur and Schmidt (1975)) is expected to eliminate most of these erroneous estimates.

The advantages of the secondary feature algorithm are more pronounced for applications in which pre-filtering is inconvenient or inappropriate. The wideband speech signal was recorded in an acoustically-deadened (but not anechoic) room. Figure 9.5 shows the secondary feature algorithm to be more reliable. It appears therefore that the latter is to be preferred whenever the signal to be processed is inherently wideband, or when speech from men, women and children is to be processed without prior selection of the pitch range.

9.7 PITCH TRAJECTORY QUANTISATION - CONVERSION TO NOTE TABLE FORM

The discussion of pitch estimation techniques has so far concentrated on the generation of the pitch trajectory in the form of pitch frequency as a function of time. Since the quantisation intervals of both time and frequency are small, it is convenient to regard the pitch trajectory as being essentially in analogue form. This section considers the conversion of the pitch trajectory into the form of a table of notes (such as the MOD data structure, cf. Section 3.4). Two distinct "quantisation" processes are required. These are the segmentation of the pitch trajectory into separate notes and the recognition of the pitch corresponding to each note. The time segmentation and pitch quantisation tasks are interrelated in the sense that both time and pitch information are often required to separate adjacent notes. For example, if a piece of music is played legato rather than staccato then successive notes are separated by a pitch discontinuity rather than by a short silence interval. Additional complications may arise if vibrato is present (since pitch variations of the order of a semitone either side of the actual note pitch can occur) or if a pitch "overshoot" occurs during the attack transient of a note. Also, the pitch quantisation procedure should include provision for notes tuned to scales other than the standard equally-tempered scale with A4 at 440 Hz. This is especially important for singing or humming, since a "cue

note" may not be available to the singer.

These considerations led to the algorithm which is now described. The pitch trajectory is sampled at 10 ms intervals, which is the same sampling rate used by the organ keyboard input system (cf. Section 3.5). Each pitch period is compared with a set of pitch period thresholds, which correspond to the time intervals midway between adjacent pitch periods of the notes of the standard equally-tempered scale. These thresholds are listed in Table 9.2, together with the frequencies and pitch periods of the equally-tempered scale tuned to A4 at 440 Hz. From the pitch period threshold examination is determined the pitch of the equally-tempered scale note which is nearest to the current pitch estimate. This "quantised" pitch is converted to the standard pitch code (listed in Table 3.3) and to the equivalent organ keyboard sample (see Section 3.5). These note parameters are stored in a short buffer memory, and pitch trajectory "smoothing" is performed. The main object of the smoothing algorithm is to delete all "notes" which are shorter than 40 ms, and to "bridge" short gaps (of up to 40 ms duration) in any note. The "smoothed" note parameters are then passed to subroutine REC (cf. Section 3.5) and the note table is constructed using the MOD data structure (cf. Section 3.3).

So far it is assumed that the note pitches are tuned accurately to the standard equally-tempered scale. However this is not generally true, and provision is made for "tuning" the system to other equally-tempered scales as follows. At the start of each pitch processing run the user

is requested to specify whether a "tune-up" is required. If so, then the first (smoothed) note whose duration exceeds one second is taken to be the tune-up reference note. The average period estimate is computed over the one second interval. The user is requested to specify whether this note is to be aligned to the nearest note of the standard equally-tempered scale or to a specified pitch (e.g. C3). From this information is computed the pitch frequency in Hz to which the reference note must be aligned. The effective signal sampling rate is then scaled so that

$$\text{DELTAT} = \text{DELTAT}' * \text{PPSPEC} / \text{PPREF} \qquad (9.46)$$

where DELTAT is the reciprocal of the effective signal sampling rate, DELTAT' is the reciprocal of the actual signal sampling rate, PPSPEC is the pitch period of the specified reference pitch, and PPREF is the averaged pitch period of the tune up note. The operators * and / denote multiplication and division, respectively. All subsequent pitch periods are computed as

$$\text{PPEST} = \text{ISAM} * \text{DELTAT} \qquad (9.47)$$

where PPEST is the estimated pitch period in seconds, ISAM is the number of signal samples in the estimated pitch period, and DELTAT is the reciprocal of the effective signal sampling rate.

The note table generated in this way is filed on disc in the same format as Piano Typewriter files.

An example which illustrates the pitch trajectory quantisation and smoothing procedure is presented in

Figure 9.6, which is the tune "Georgy Girl" by T. Springfield, as hummed by Susan Frykberg. Figure 9.6(a) shows the pitch trajectory obtained using the third software implementation of the Gold and Rabiner algorithm (see Section 9.3). The quantised pitch trajectory is displayed in MOD notation in Figure 9.6(b). The poor pitch frequency resolution of the pitch estimator (which results from the low signal sampling rate of about 8.3 kHz) is apparent when Figures 9.6(a) and (b) are compared. Nevertheless subsequent "smoothing" (*not* manual editing) of the MOD note table results in the display shown in Figure 9.6(c). The note-table smoothing algorithm used here deletes all notes whose duration is less than 100 ms. Also, if two or more notes overlap, all but the longest note in each overlapping note cluster is deleted. Figure 9.6(d) illustrates the TRAD notation display produced from the smoothed MOD note table using the transcription and editing system described in Chapter 4.

9.8 SUMMARY AND CONCLUSIONS

This chapter considers in detail the use of pitch estimation techniques which operate in the time domain by recognising recurring features of the signal waveshape. A general discussion of such methods is presented (cf. Section 9.2). This discussion forms a unified basis for the various algorithms described in Section 7.8, and suggests how these algorithms can be extended to include more than one "feature" of the signal waveshape. The well-known Gold and Rabiner (1969) algorithm is next described,

together with several modifications which extend its pitch range to six octaves. The problems encountered during the implementation of this algorithm are identified, and various hardware-software tradeoffs are considered to speed the processing and to facilitate the temporary storage of the signal.

While the Gold and Rabiner algorithm works well for many speech and music signals, it fails for those signals which possess several peaks of similar magnitude in each pitch period. A new algorithm which is designed to operate successfully on such signals is described in Section 9.4. This new algorithm possesses a simpler logical structure than the Gold and Rabiner algorithm, so that the implementation in hardware or software is less complicated for the former than for the latter. Another advantage of the new algorithm is that it is analytically relatable to autocorrelation analysis (cf. Section 9.5). The results presented in Section 9.6 show that for wideband speech and music signals the new algorithm performs better than the Gold and Rabiner algorithm.

Future work should be directed towards real-time implementation. The first priority is the development of dedicated hardware which measures in real time the signal "primary features" (cf. Section 9.2) and their times of occurrence, using a signal sampling rate of at least 50 kHz. This hardware could be incorporated directly into the signal preprocessor and magnetic tape storage facility used in the third software implementation of the Gold and Rabiner algorithm (see Section 9.3). In this way would be overcome

the poor pitch frequency resolution which at present limts the usefulness of the latter. The second priority is to refine the new algorithm, since no attempt has so far been made to "optimise" it for computational efficiency. For example, the need for a divide operation in equation (9.27) should be examined in the light of the comments in the paragraph which contains equation (9.21). The conversion of the existing program from floating-point to fixed-point arithmetic should also be considered, and the appropriate word size and scaling factors (if any) should be established. The memory stack lengths should also be investigated further, although those used in the existing program seem suitable. The resulting "optimised" program could also be incorporated with the signal preprocessor and magnetic tape storage facility to provide a useful fast (but not real-time) pitch estimator which overcomes many of the limitations of the Gold and Rabiner system. Finally, the design and construction of dedicated hardware which achieves real-time operation should be undertaken. Continuing developments in microprocessor technology will undoubtedly influence the design - for example it could soon be economically feasible to interconnect several 16-bit machines to achieve real-time operation.

TABLE 9.1

Summary of results from representative musical instrument signals. Each signal analysed is a 75 ms portion of a steady note. The pitch estimate $1/\tilde{T}$ is computed at 5 ms intervals. The improved algorithm employs pitch averaging (equation 9.28) over a 10 ms interval for these results (the usual averaging interval is 25 ms which further reduces the standard deviation). Gross errors in $\tilde{T}$ are defined as those estimates which differ by more than 10% from the eye-detected period. Gross errors are not included in the mean and standard deviation calculations.

INSTRUMENT	GOLD AND RABINER ALGORITHM			IMPROVED ALGORITHM		
	Mean (Hz)	Standard Deviation (Hz)	No. of Gross Errors in T	Mean (Hz)	Standard Deviation (Hz)	No. of Gross Errors in T
OBOE						
Note 1	557.4	2.9	0	556.0	0.0	0
" 2	556.1	3.6	0	563.0	0.0	0
" 3	561.5	2.9	0	563.0	0.0	0
" 4	558.3	3.4	0	556.0	0.0	0
SAXOPHONE						
Note 1	299.1	1.6	0	298.5	1.1	0
" 2	301.0	1.4	0	300.8	0.6	0
" 3	300.3	1.6	0	299.3	0.8	1
" 4	301.2	1.0	0	301.0	0.0	0
VIOLIN						
Note 1	462.3	4.1	0	460.4	1.4	0
" 2	462.7	3.3	2	460.4	8.2	0
" 3	463.1	3.8	0	462.5	4.0	0
" 4	457.1	3.2	0	455.7	1.8	0

TABLE 9.1 (Continued)

FRENCH HORN						
Note 1	332.2	2.9	0	331.2	2.5	0
" 2	324.3	3.0	0	324.7	0.7	0
" 3	327.2	2.7	1	326.1	1.5	0
" 4	378.1	6.2	0	379.5	6.3	0
TRUMPET						
Note 1	414.3	2.6	0	414.5	2.6	0
" 2	408.0	0.0	11	405.1	4.3	0
" 3	417.0	0.0	11	412.0	0.0	0
" 4	410.9	1.8	0	408.9	1.7	0
FLUTE						
Note 1	852.4	11.0	0	851.0	0.0	0
" 2	853.6	9.6	0	851.0	0.0	0
" 3	853.5	6.7	0	851.0	0.0	0
BASSOON						
Note 1	183.1	0.4	0	183.0	0.0	0
" 2	186.2	1.3	1	185.3	0.6	0
" 3	185.9	0.6	0	185.9	0.4	0
CLARINET						
Note 1	331.9	1.0	0	331.3	0.7	0
" 2	330.6	1.1	0	329.9	1.5	0
" 3	332.2	1.5	0	331.0	0.0	0
TOTAL GROSS ERRORS			26			1

TABLE 9.2

FREQUENCIES AND PITCH PERIODS OF THE EQUALLY-TEMPERED SCALE.

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C9	8372.01	119.		
			8137.07	123.
B8	7902.13	127.		
			7680.37	130.
A#8	7458.62	134.		
			7249.30	138.
A8	7040.00	142.		
			6842.43	146.
G#8	6644.87	150.		
			6458.40	155.
G8	6271.92	159.		
			6095.91	164.
F#8	5919.91	169.		
			5753.78	174.
F8	5587.65	179.		
			5430.84	184.
E8	5274.04	190.		
			5126.03	195.
D#8	4978.03	201.		
			4838.33	207.
D8	4698.63	213.		
			4566.77	219.
C#8	4434.92	225.		
			4310.46	232.
C8	4186.00	239.		

TABLE 9.2 (Continued 2)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C8	4186.00	239.		
			4068.53	246.
B7	3951.06	253.		
			3840.18	260.
A#7	3729.31	268.		
			3624.65	276.
A7	3520.00	284.		
			3421.21	292.
G#7	3322.43	301.		
			3229.20	310.
G7	3135.96	319.		
			3047.95	328.
F#7	2959.95	338.		
			2876.89	348.
F7	2793.82	358.		
			2715.42	368.
E7	2637.02	379.		
			2563.01	390.
D#7	2489.01	402.		
			2419.16	413.
D7	2349.31	426.		
			2283.38	438.
C#7	2217.46	451.		
			2155.23	464.
C7	2093.00	478.		

TABLE 9.2 (Continued 3)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C7	2093.00	478.		
			2034.26	492.
B6	1975.53	506.		
			1920.09	521.
A#6	1864.65	536.		
			1812.32	552.
A6	1760.00	568.		
			1710.60	585.
G#6	1661.21	602.		
			1614.60	619.
G6	1567.98	638.		
			1523.97	656.
F#6	1479.97	676.		
			1438.44	695.
F6	1396.91	716.		
			1357.71	737.
E6	1318.51	758.		
			1281.50	780.
D#6	1244.50	804.		
			1209.58	827.
D6	1174.65	851.		
			1141.69	876.
C#6	1108.73	902.		
			1077.61	928.
C6	1046.50	956.		

TABLE 9.2 (Continued 4)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C6	1046.50	956.		
			1017.13	983.
B5	987.76	1012.		
			960.04	1042.
A#5	932.32	1073.		
			906.16	1104.
A5	880.00	1136.		
			855.30	1169.
G#5	830.60	1204.		
			807.30	1239.
G5	783.99	1276.		
			761.98	1312.
F#5	739.98	1351.		
			719.22	1390.
F5	698.45	1432.		
			678.85	1473.
E5	659.25	1517.		
			640.75	1561.
D#5	622.25	1607.		
			604.79	1653.
D5	587.32	1703.		
			570.84	1752.
C#5	554.36	1804.		
			538.80	1856.
C5	523.25	1911.		

TABLE 9.2 (Continued 5)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C5	523.25	1911.		
			508.56	1966.
B4	493.88	2025.		
			480.02	2083.
A#4	466.16	2145.		
			453.08	2207.
A4	440.00	2273.		
			427.65	2338.
G#4	415.30	2408.		
			403.65	2477.
G4	391.99	2551.		
			380.99	2625.
F#4	369.99	2703.		
			359.61	2781.
F4	349.22	2863.		
			339.42	2946.
E4	329.62	3034.		
			320.37	3121.
D#4	311.12	3214.		
			302.39	3307.
D4	293.66	3405.		
			285.42	3504.
C#4	277.18	3608.		
			269.40	3712.
C4	261.62	3822.		

TABLE 9.2 (Continued 6)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C4	261.62	3822.		
			254.28	3933.
B3	246.94	4050.		
			240.01	4166.
A#3	233.08	4290.		
			226.54	4414.
A3	220.00	4545.		
			213.82	4677.
G#3	207.65	4816.		
			201.82	4955.
G3	195.99	5102.		
			190.49	5249.
F#3	184.99	5405.		
			179.80	5562.
F3	174.61	5727.		
			169.71	5892.
E3	164.81	6067.		
			160.18	6243.
D#3	155.56	6428.		
			151.19	6614.
D3	146.83	6810.		
			142.71	7007.
C#3	138.59	7215.		
			134.70	7424.
C3	130.81	7645.		

TABLE 9.2 (Continued 7)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C3	130.81	7645.		
			127.14	7865.
B2	123.47	8099.		
			120.00	8333.
A#2	116.54	8581.		
			113.27	8828.
A2	110.00	9091.		
			106.91	9353.
G#2	103.82	9631.		
			100.91	9910.
G2	97.99	10204.		
			95.24	10499.
F#2	92.49	10811.		
			89.90	11123.
F2	87.30	11454.		
			84.85	11785.
E2	82.40	12135.		
			80.09	12485.
D#2	77.78	12856.		
			75.59	13228.
D2	73.41	13621.		
			71.35	14014.
C#2	69.29	14431.		
			67.35	14848.
C2	65.40	15289.		

TABLE 9.2 (Continued 8)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C2	65.40	15289.		
			63.57	15730.
B1	61.73	16198.		
			60.00	16666.
A#1	58.27	17161.		
			56.63	17657.
A1	55.00	18182.		
			53.45	18707.
G#1	51.91	19263.		
			50.45	19819.
G1	48.99	20408.		
			47.62	20998.
F#1	46.24	21622.		
			44.95	22246.
F1	43.65	22908.		
			42.42	23569.
E1	41.20	24270.		
			40.04	24971.
D#1	38.89	25713.		
			37.79	26455.
D1	36.70	27242.		
			35.67	28029.
C#1	34.64	28862.		
			33.67	29695.
C1	32.70	30578.		

TABLE 9.2 (Continued 9)

NOTE	FREQUENCY (HZ)	PITCH PERIOD (MICROSEC.)	DECISION THRESHOLD (HZ)	DECISION THRESHOLD (MICROSEC.)
C1	32.70	30578.		
			31.78	31461.
B0	30.86	32396.		
			30.00	33332.
A#0	29.13	34323.		
			28.31	35314.
A0	27.50	36364.		
			26.72	37414.
G#0	25.95	38526.		
			25.22	39638.
G0	24.49	40817.		
			23.81	41995.
F#0	23.12	43244.		
			22.47	44492.
F0	21.82	45815.		
			21.21	47138.
E0	20.60	48540.		
			20.02	49941.
D#0	19.44	51426.		
			18.89	52911.
D0	18.35	54484.		
			17.83	56057.
C#0	17.32	57724.		
			16.83	59390.
C0	16.35	61156.		

TABLE 9.3

Values of $\varepsilon_3(M_{1,i})$ used in the modified Gold and Rabiner algorithm (see equation 9.21). All values are in ms.

$M_{1,i}$	$\varepsilon_3(M_{1,i})$
0.4 to 0.8	0.03
0.8 to 1.6	0.05
1.6 to 3.1	0.1
3.1 to 6.3	0.2
6.3 to 12.7	0.4
12.7 to 25.5	0.8

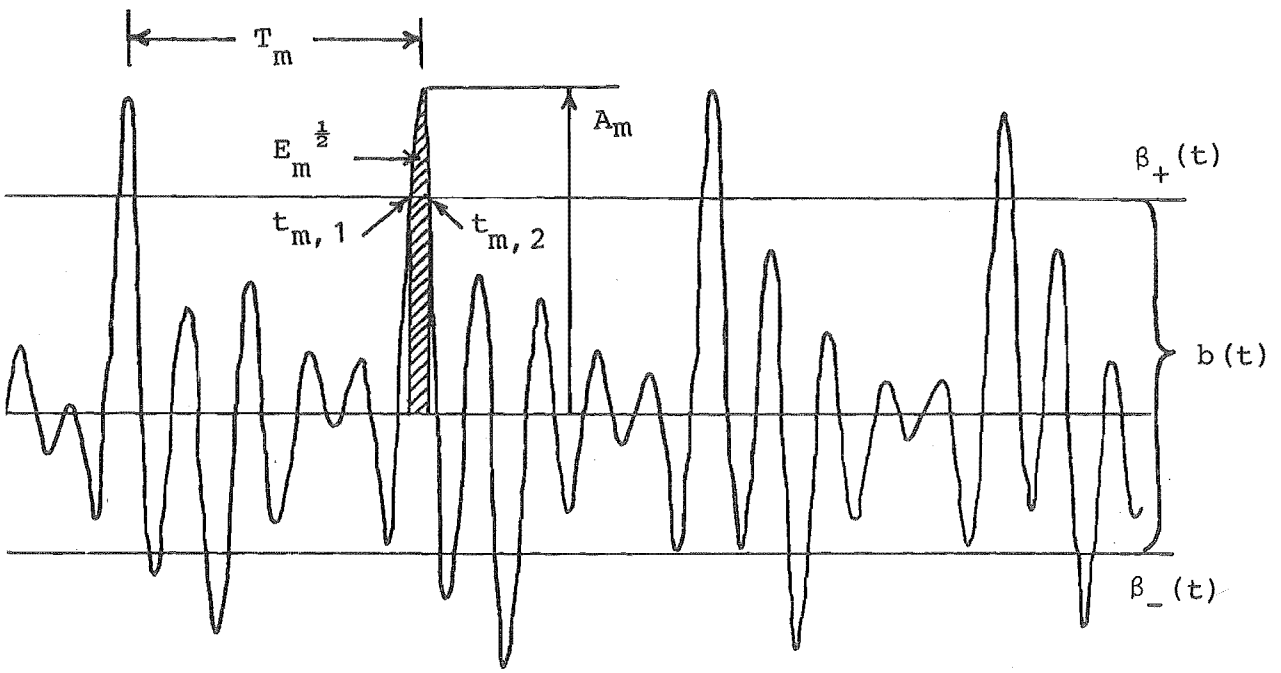
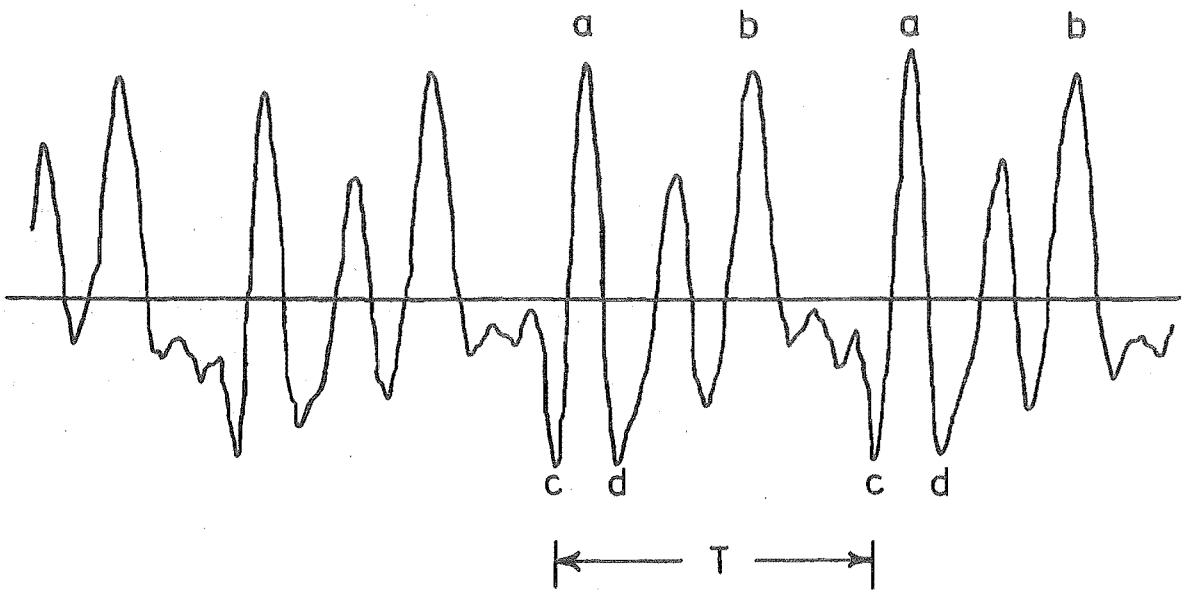
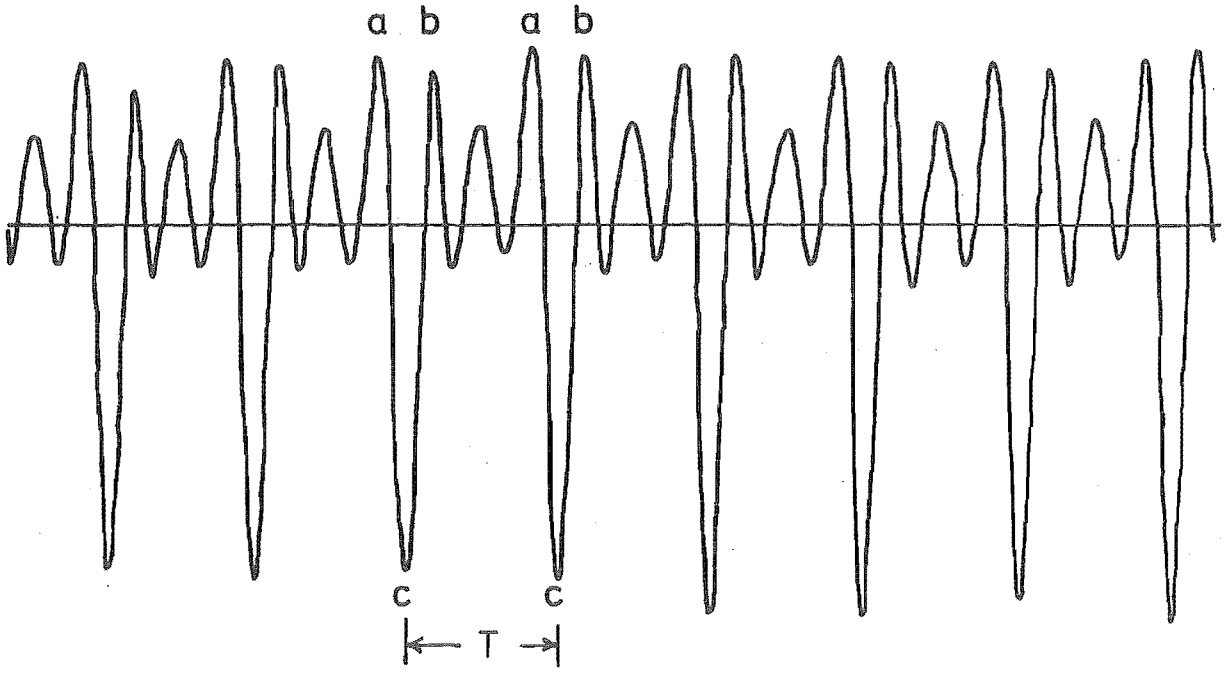


Figure 9.1 Illustrating the definitions of signal thresholds and primary features. The signal shown is speech (a segment of the vowel /ʌ/, male speaker) after filtering. The pass band is 100 Hz to 600 Hz.



(a)



(b)

Figure 9.2 Illustrating signals which possess multiple peaks of similar amplitude within each pitch period.

(a) Speech. A segment of the vowel /a/, male speaker.

(b) Trumpet. Waveforms of this kind cause the Gold and Rabiner algorithm to fail (see notes 2 and 3 of Table 9.1).

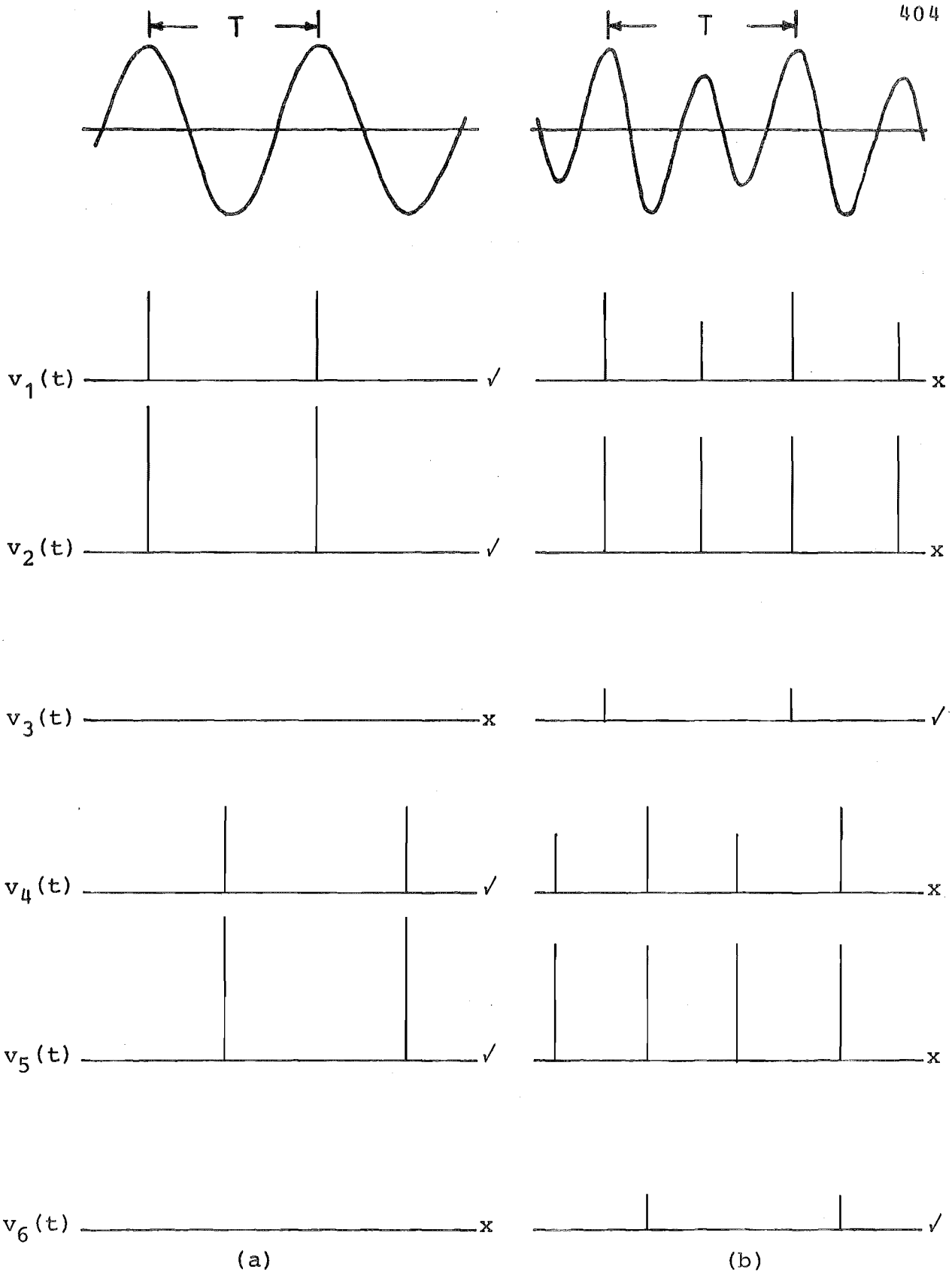


Figure 9.3 Two examples which illustrate the six scalar functions $v_i(t)$ used by the six independent period estimators.

(a) Waveform with fundamental component only. Period estimates 1, 2, 4 and 5 are correct, while 3 and 6 are incorrect.

(b) Waveform with fundamental and strong second harmonic component. Period estimates 3 and 6 are correct, while 1, 2, 4 and 5 are incorrect.

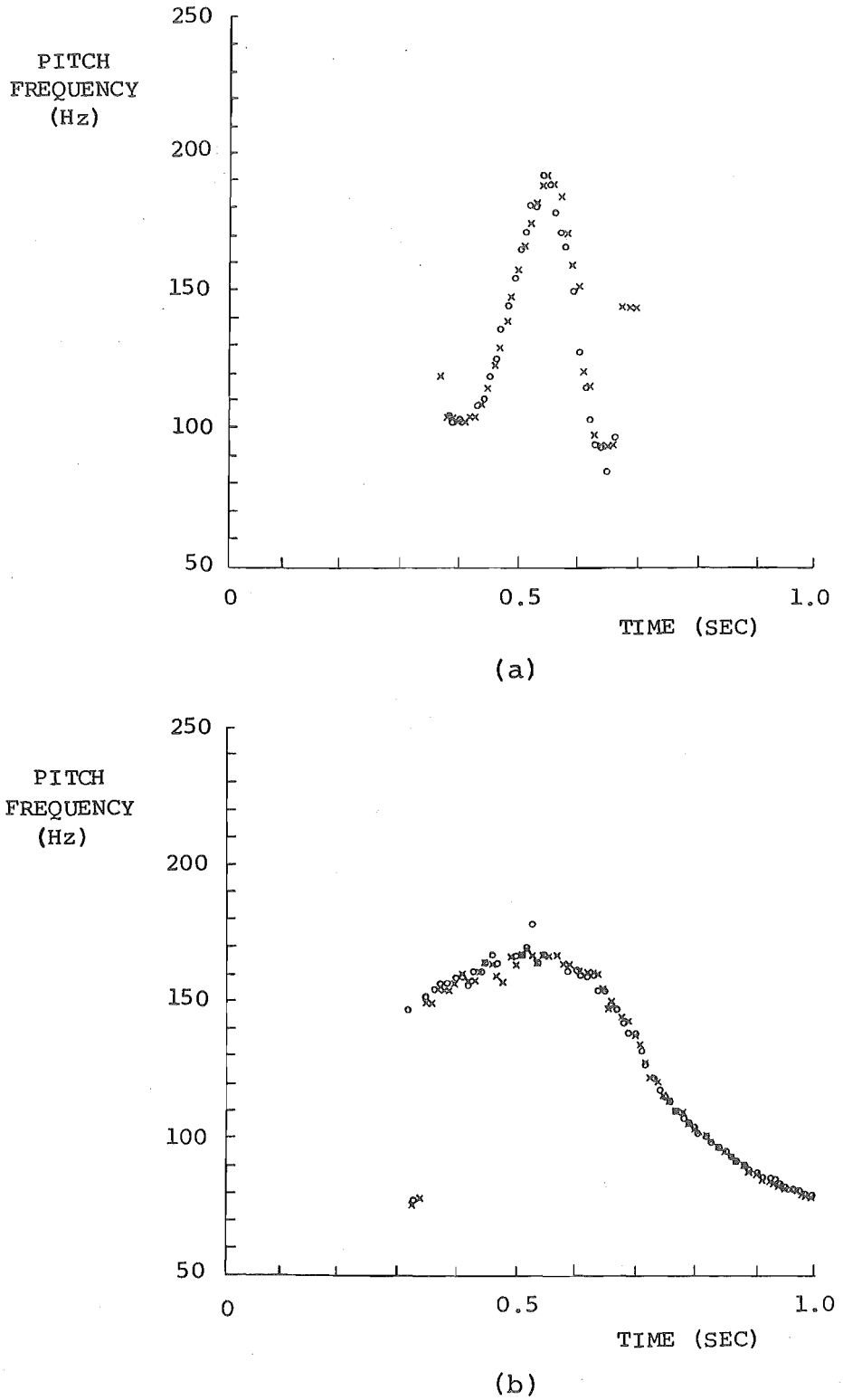
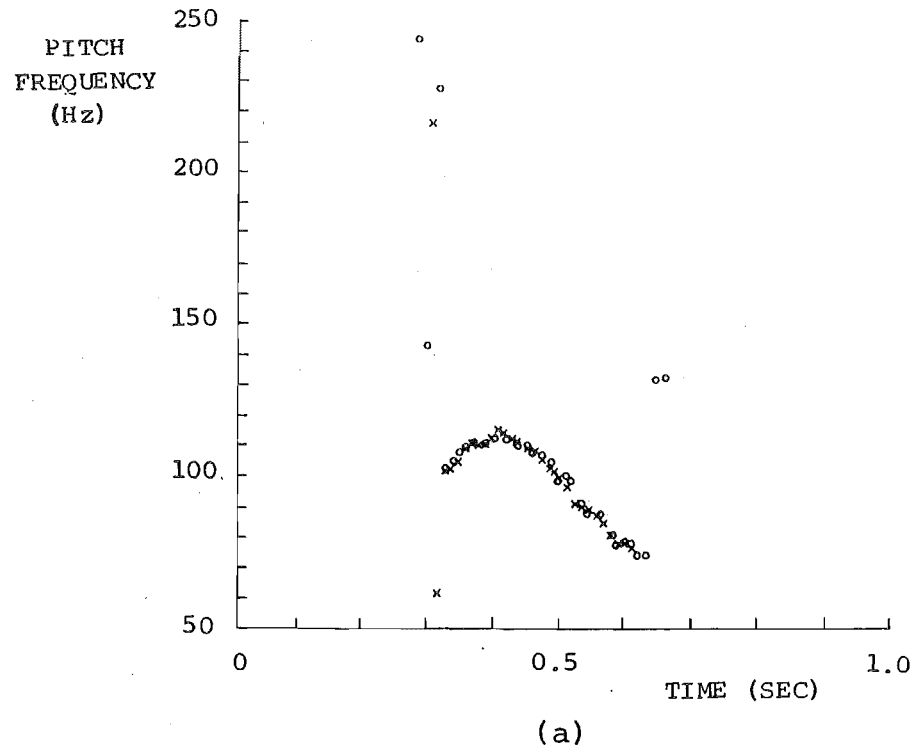


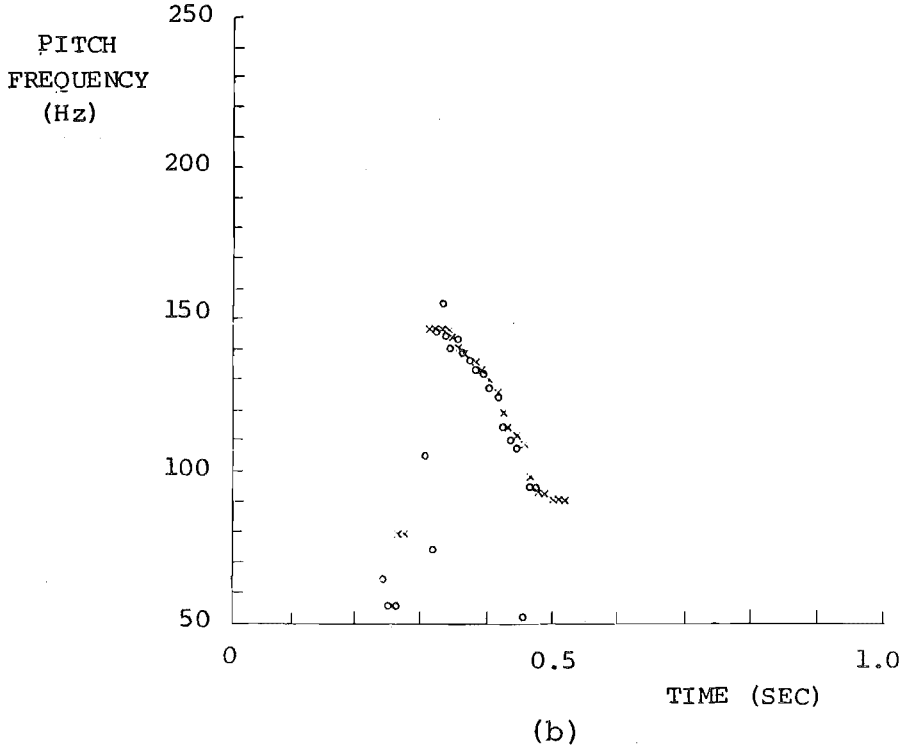
Figure 9.4 Pitch trajectories for prefiltered utterances of a male speaker.

(a) "yes"
(b) "no"

xxxx Improved algorithm
oooo Gold and Rabiner algorithm.



(a)



(b)

Figure 9.5 Pitch trajectories for wideband utterances of a male speaker.

(a) "one"
(b) "two"

xxxx Improved algorithm
oooo Gold and Rabiner algorithm

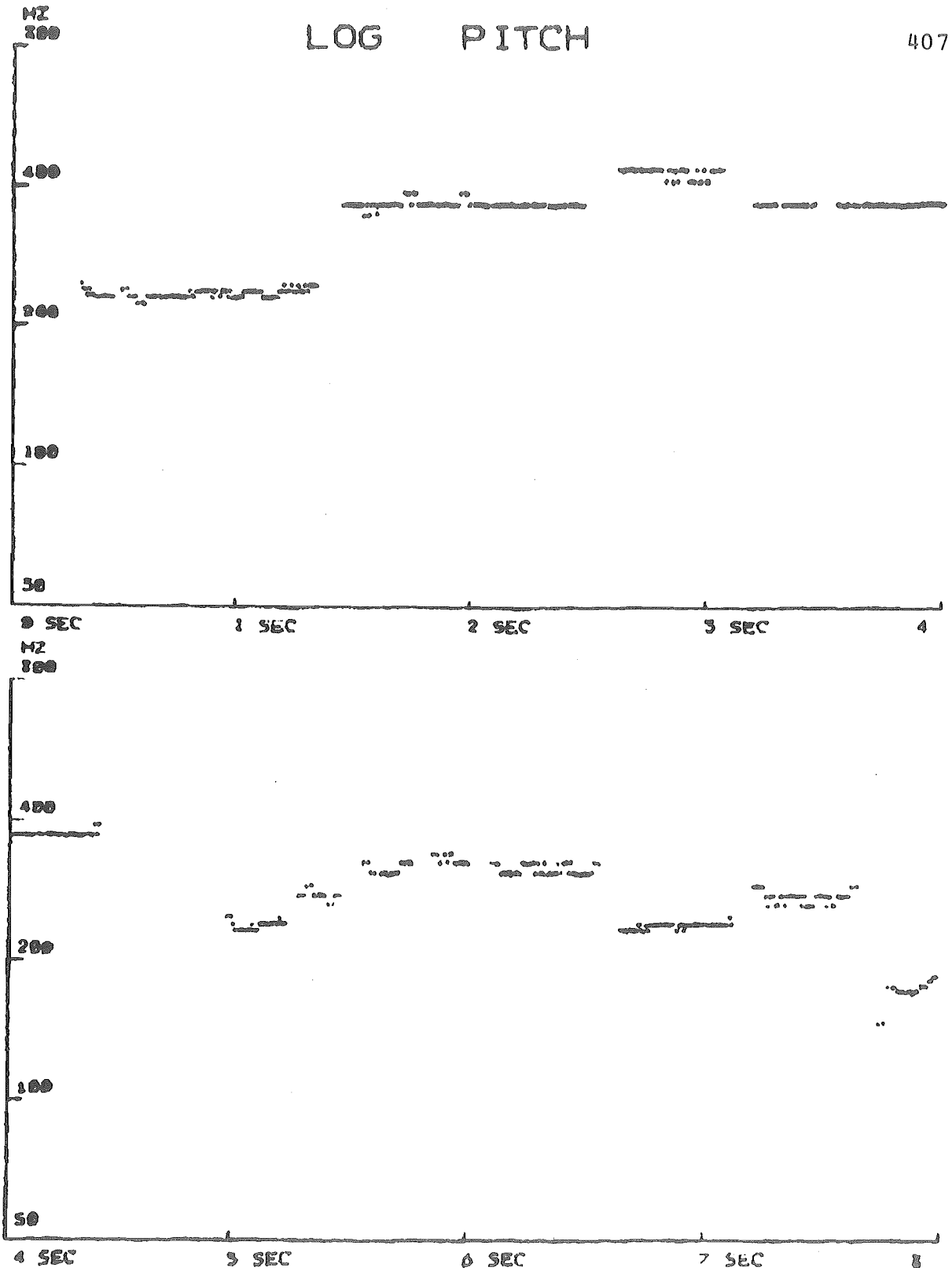


Figure 9.6 Example of pitch trajectory quantisation and display in MOD and TRAD notations (cf. Chapters 3 and 4). The piece shown is "Georgy Girl" (T. Springfield) as hummed by Susan Frykberg in a quiet room.

(a) Original pitch trajectory (2 pages). The relatively poor pitch frequency resolution of the present implementation is apparent (see Section 9.3).

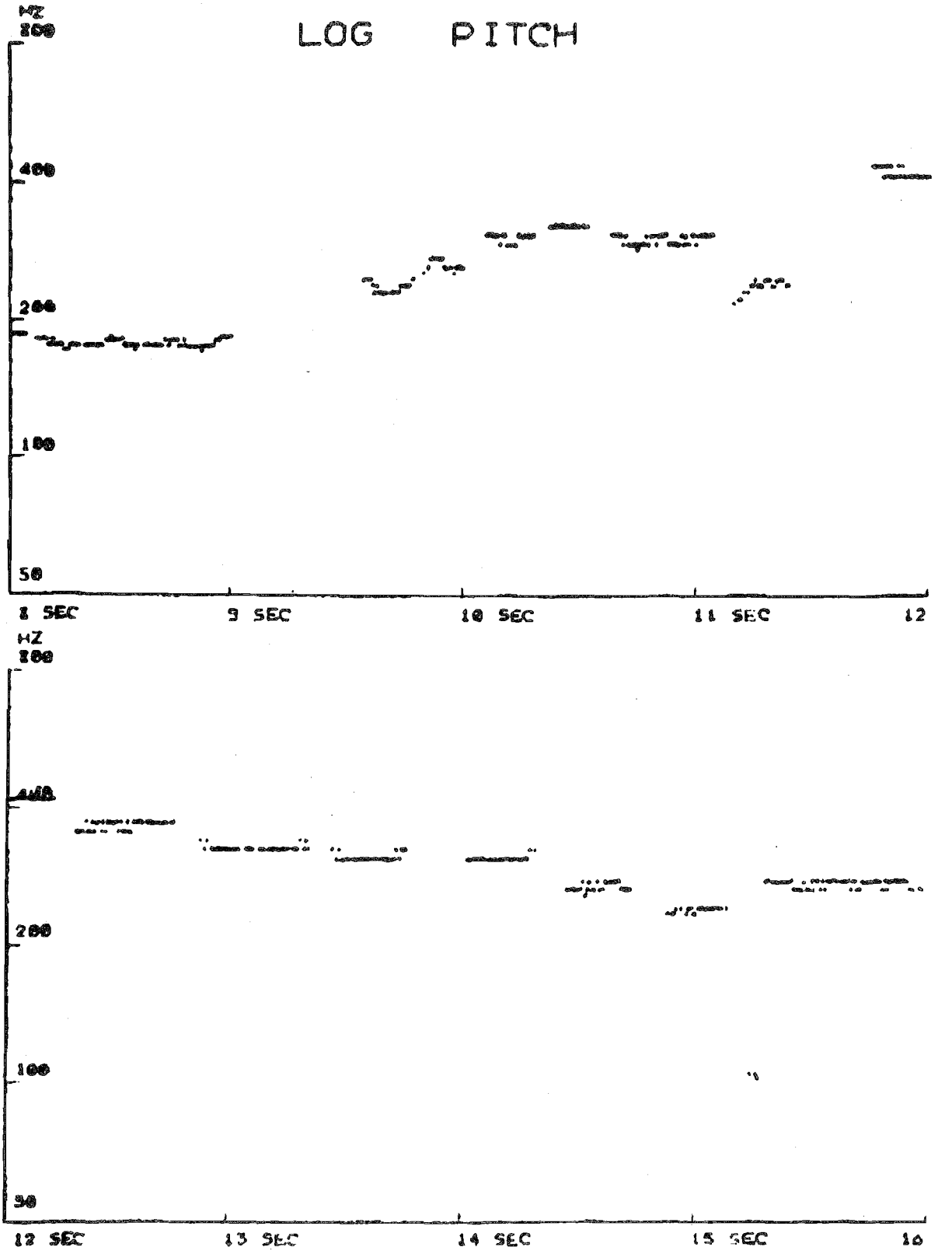


Figure 9.6 (a) Continued.

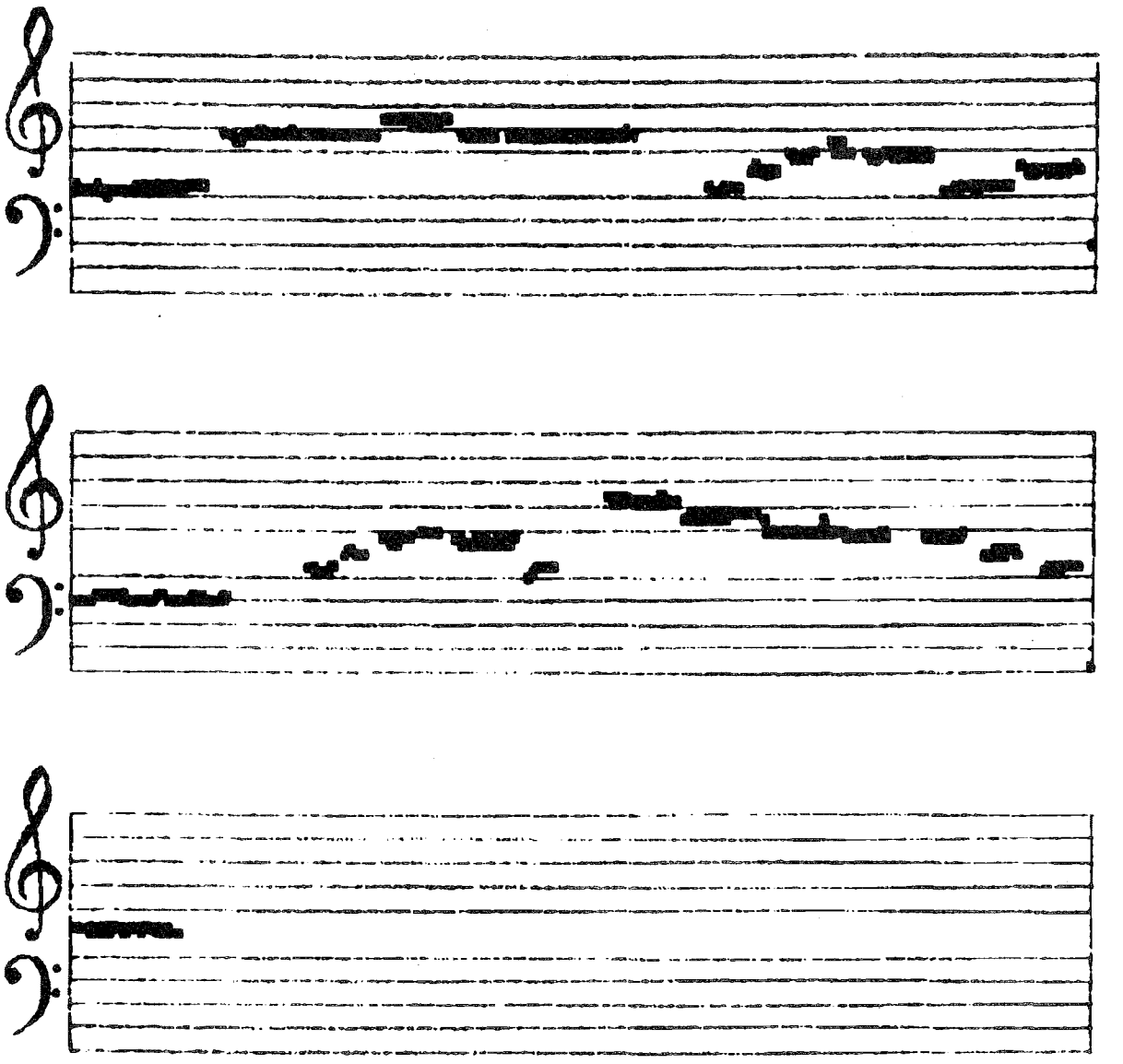


Figure 9.6 (b) Display of unsmoothed quantised pitch trajectory in MOD notation.

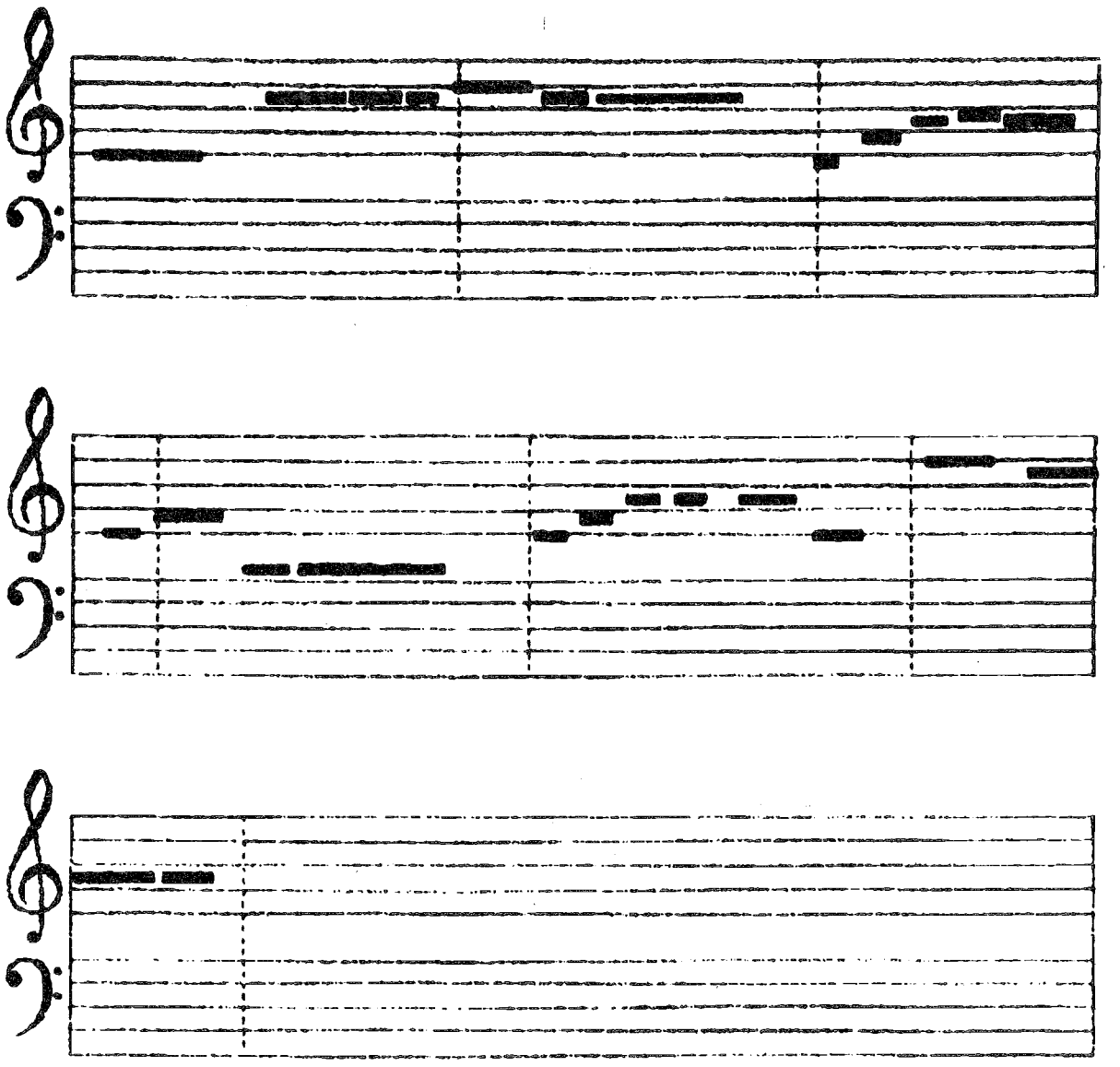


Figure 9.6 (c) Display of smoothed quantised pitch trajectory. No manual editing has been done.

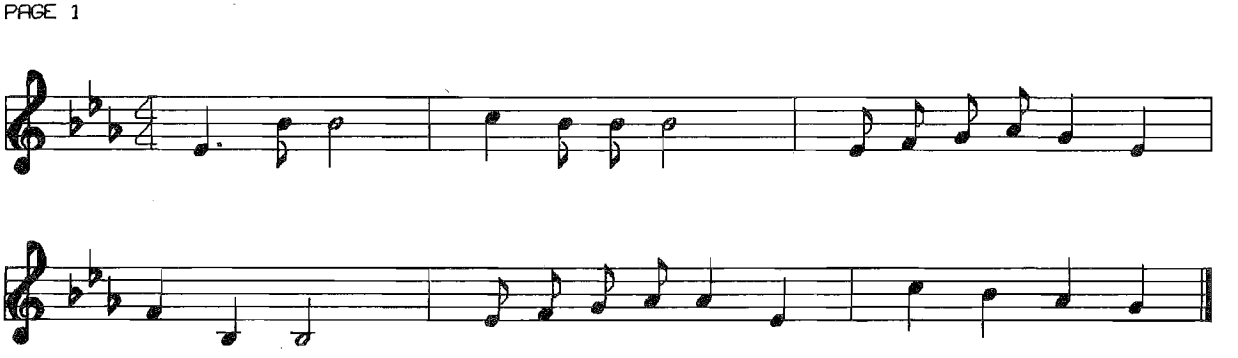


Figure 9.6 (d) Display in TRAD notation of the passage shown in Figure 9.6(c) after transcription and editing. The music notation used is identical to that of the original score.

PART 4

SUMMARY AND CONCLUSIONS

Most writers regard truth as their most valuable possession, and therefore are most economical in its use.

Mark Twain.

CHAPTER 10

SUMMARY AND CONCLUSIONS

The central theme of this thesis is the application of digital computing techniques to music. Particular emphasis is placed on interactive systems in which the computer serves as a "secretary" or "assistant" to the creative human. Thus the underlying philosophy is that the computer should complement rather than supplant the musician, composer, or music teacher. Accordingly, considerable attention is paid to the design of the man-machine interface and to the organisation of a system which provides a wide range of facilities. Other topics which are discussed in detail include the use of digital techniques for sound synthesis and the estimation of the pitch trajectories of speech and music signals.

10.1 COMPUTER AIDS FOR MUSICIANS

The early applications of computers to music composition, music performance, and music printing operated in "batch" rather than "interactive" mode. By the early 1970's it became widely recognised that interactive systems offer substantial advantages over systems which operate in batch processing mode. However, to achieve their potential usefulness, interactive systems require that the man-machine interface be designed so that the user approaches the system

with a sense of convenience. Thus the user should interact with the system using aural and visual media that are already familiar, rather than be forced to learn an alphanumerical "language" which describes sounded or written music. Similarly, the commands used to specify the desired tasks should closely resemble the terminology normally used to describe those tasks. The system described in Chapters 3 and 4 was designed with these goals as the primary objective. This system is used for teaching, composing, and producing music typescript. In addition it is a useful interface for other computer music applications, and is used by Lamb (1977) and Susan Frykberg as a general-purpose "music operating system".

The automatic transcription of aural music into the corresponding written conventional notation has long been desired by both musicians and composers (see for example Ashton, 1970). Fully automatic transcription is fraught with difficulties, and Kassler (1977) has concluded that this approach is not suitable for score preparation in a commercial music printing environment. Nevertheless the transcription, display and editing system described in Chapter 4 overcomes many of the difficulties mentioned by Kassler. The system incorporates fast, flexible editing facilities which permit a human operator to manually correct any errors perpetrated by the transcription system. The score of an original composition by P. Norman has been produced in this way by the composer (who is not a trained computer operator). Specific suggestions which should improve the speed and flexibility of the system are made

in Chapter 4.

10.2 SOUND SYNTHESIS

The ability to generate and control sounds using electronic techniques has created new areas for musical exploration. Not only may new sounds be generated, but the traditional limitations of manual dexterity may be removed from musical performance. The methods of electronic sound synthesis are reviewed in Chapter 5, with particular emphasis on digital techniques. While this thesis was in press, the extensive review paper by Moorer (1977) appeared. Moorer's review parallels much of the material presented in the last two sections of Chapter 5. He also discusses the use of linear-prediction for music synthesis, and cites work by Peterson (1975, 1976a, 1976b) of which the author was unaware. Consequently the comments in the penultimate paragraph of Section 5.4 should be revised to acknowledge Petersen's research.

Many existing systems which use digital techniques for sound synthesis are implemented using a computer to produce the samples which are subsequently converted to analogue form. It is more appropriate to delegate this task to dedicated hardware, freeing the computer to perform supervisory control. In this way real-time interaction is possible, and the human operator can "orchestrate" or "conduct" the performance. An additional advantage of this approach is that nuances of performance can be controlled in real-time, rather than be required to be predetermined. A digital synthesis system which incorporates these ideas is

described in Chapter 6. This system is being developed as a continuing series of M.E. projects, and a great deal of further work is required before its full potential is realised. In particular, the development of interactive control software is required so that performances may be more conveniently constructed.

10.3 PITCH TRAJECTORY ESTIMATION

Analysis-synthesis telephony provided a strong motivation for the development of techniques for pitch estimation from speech signals. The difficulty of this problem is attested to by the large number of methods which have been developed (McKinney, 1965; Rabiner, Cheng, Rosenberg and McGonegal, 1976). Many of the recently developed techniques utilise signal characteristics which are specific to speech. In particular, cepstrum and inverse filtering analysis work well for speech but fail consistently for signals produced by some musical instruments.

An extensive review of pitch estimation methods is given in Chapter 7. The applicability of each method to signals other than speech is assessed, and an indication is given of the amount of computation required. This latter is particularly important when signals whose pitches span more than two or three octaves are to be analysed. For example, in many cases the computation cost increases as the square of the signal sampling rate, which in turn depends upon the pitch range. The interdependence between sampling rate, pitch range, and pitch frequency resolution

is analysed for discrete autocorrelation pitch estimation. It is concluded that autocorrelation analysis is the most universally applicable pitch estimation method, although signal preprocessing (such as adaptive centre-clipping) should be used for speech to reduce the effect of the vocal tract response. The main disadvantage of autocorrelation analysis is that it is computationally expensive. Various computational techniques which use number theoretic transforms and Walsh transforms to efficiently evaluate the autocorrelation function are reviewed in Chapter 8. While these numerical techniques offer some advantages over the conventional FFT method, the decrease in computation effort is not sufficient to make autocorrelation analysis attractive for fast, inexpensive pitch estimation. A more suitable approach is to attempt to recognise recurring features of the time-domain signal waveform.

In Chapter 9 a general basis of time-domain feature-recognition pitch estimation methods is given. The well-known Gold and Rabiner (1969) algorithm is described, and the problems encountered in its implementation are identified. The Gold and Rabiner algorithm uses only the amplitude of the signal maxima and minima to determine the signal periodicity. Consequently it is unsuccessful for signals which contain several peaks of similar amplitude in each pitch period. A new algorithm is presented which overcomes this deficiency by using additional "features" of the signal. Results from both speech and musical signals show that the new algorithm works over a wider class of signals than does the Gold and Rabiner

algorithm. In addition, the new algorithm possesses a simple logical structure and is analytically relatable to autocorrelation pitch estimation. Specific suggestions for the development of both hardware and software to achieve real-time operation are made in Section 9.8.

APPENDIX

A SIGNAL DATA ACQUISITION, DISPLAY AND EDITING SYSTEM FOR THE EAI 590 HYBRID COMPUTER

This appendix describes a system which permits signals to be sampled, stored, and subsequently edited using an interactive display. The present system is an extension of A.B. Robson's signal acquisition system, and was developed specifically for use in conjunction with a Fourier and Walsh spectral analysis software package developed by the author. This latter was used to produce the results presented in Chapters 7 and 8. The use of the signal acquisition and storage portion of the system for pitch estimation is described in Chapter 9.

The computing facilities available within the University dictated the form of the system. The most convenient of these is the Electrical Engineering Department's EAI 590 hybrid computer, which consists of an EAI 640 digital computer with 16 K 16 bit words of magnetic core memory and fixed head disk storage for an additional 360 K words, a small analogue computer (the EAI 580), and a hybrid interface between the two units. Two-way communication between the analogue and digital computers is provided by four logic sense lines, four logic control lines, two general-purpose interrupt lines, 16 ADC channels, and six DAM channels. The maximum ADC conversion

rate is approximately 42 kHz. An interactive graphics facility is incorporated with the digital computer - this is described in Section 3.2. Supplementary digital storage is provided by magnetic tape (150 K words on each tape, with a maximum word-write rate of about 5.0 kHz) and paper tape. Direct memory access is not incorporated.

The limitation of this computing configuration for high-speed acquisition and storage of signals whose durations span more than a few seconds is discussed in detail in Section 9.3, where the use of both analogue and digital magnetic tape as well as disc storage is considered. It transpires that for signal sampling rates of more than about 10 kHz an intermediate storage buffer in core memory is required. Since no DMAC is provided, this buffered signal cannot be written on to disc without interrupting the signal sampling procedure for an unacceptably long interval (typically 20 to 40 ms). The construction of a dedicated high-speed memory buffer connected between the ADC and the CPU to overcome this problem was investigated. However the cost of development of such hardware was not considered to be justified. As a consequence of this decision the software system described below was developed.

The signal acquisition module operates as follows. The analogue signal to be sampled is patched into ADC channel 0 (which incorporates an analogue sample-and-hold to minimise the errors incurred by the non-zero conversion time). A clock which produces pulses at the required signal sampling rate is patched into general-purpose interrupt line 0. Logic circuitry is provided so that a logic HI

signal into sense-line 3 initiates the sampling procedure. Normally this logic HI is generated when a "start button" is pushed.

Once the sampling procedure is initiated the signal is sampled each time the sample clock interrupt occurs. If this sample interrupt rate exceeds 42 kHz then the latter sampling rate is used. The signal samples (digitised to 14 bits) are stored one sample per word in a 12 K word core buffer (the remaining 4 K words of core are required by the sampling software, which is efficiently written in ASSEMBLY language). Thus a total signal duration of 1.2 seconds may be stored if the sampling rate is 10 kHz. An alternative option (selected by depressing sense switch B) truncates each signal sample to 8 bits and packs two samples into each word. However this option is of limited usefulness because of the consequential reduction in signal dynamic range. The incorporation of a non-linear (e.g. logarithmic) encoding characteristic would reduce this problem. However the additional processing time and memory required to implement a non-linear encoding characteristic in software (e.g. using table look-up) could nullify its advantages.

When the core buffer is full the signal samples are automatically displayed, and if required are written on to a disc file, together with identifying comments.

The display and edit module reads the specified data file, requests the user to specify the signal sampling rate used, and displays the signal frame by frame. Each display frame is scaled to occupy 20 ms. Above the signal is written the frame sample number which corresponds to each

zero-crossing, local maximum, and local minimum. These sample numbers permit the user to specify the beginning and end of a "data frame" which is subsequently written on to a COMMON array for use by other core-image phases. An elementary disc operating system is incorporated within the display and edit module interpreter so that these processing phases can be conveniently executed. It is worth commenting that the signal display module can also be used for manual estimation of pitch period "by eye" (cf. McGonegal, Rabiner and Rosenberg, 1975).

REFERENCES

Abberton E. (1972). Visual feedback and intonation learning. Proc. Seventh Int. Congress of Phonetic Sciences, Mouton, The Hague, Paris.

Abberton E. and Fourcin A.J. (1973). A visual display for teaching intonation and rhythm. English Language Teaching Documents, Number 73/5.

Agarwal R.C. and Burrus C.S. (1974). Fast convolution using Fermat number transforms with applications to digital filtering. IEEE Trans. Acoust., Speech, Signal Processing., vol. ASSP-22, pp. 87-97.

Agarwal R.C. and Burrus C.S. (1975). Number theoretic transforms to implement fast digital convolution. Proc. IEEE, vol. 63, pp. 550-560.

Ahmed N., Abdussattar A.L. and Rao K.R. (1972). Efficient computation of the Walsh-Hadamard transform spectral modes. Proc. Symp. Applications of Walsh Functions, pp. 276-279.

Allen J. (1975). Computer architecture for signal processing. Proc. IEEE, vol. 63, pp. 624-633.

Alonso S., Appleton J.H. and Jones C. (1975). A special purpose digital system for the instruction, composition and performance of music. Proc. Sixth Conference on Computers in the Undergraduate Curriculum, Washington State University, pp. 17-22.

Ananthapadmanabha T.V. and Yegnanarayana B. (1975). Epoch extraction of voiced speech. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 562-570.

Anderson F. (1960). An experimental pitch indicator for training deaf scholars. Journ. Acoust. Soc. Amer., vol. 32, pp. 1065-1074.

Arenson M. (1975). A proposal for the implementation of computer managed instruction in conjunction with a basic music theory course at Iowa State University. Dept. of Music, Iowa State University. Cited by Hofstetter (1976).

Ashton A.C. (1970). Electronics, Music and Computers. Ph.D. Dissertation, University of Utah, Salt Lake City.

Atal B.S. (1968). Automatic Speaker Recognition Based on Pitch Contours. Ph.D. Thesis, Polytech. Inst. of Brooklyn.

Atal B.S. and Schroeder M.R. (1970). Adaptive predictive coding of speech signals. Bell Syst. Tech. Journ., vol. 49(8), pp. 1973-1986.

Atal B.S. and Hanauer S.L. (1971). Speech analysis and synthesis by linear prediction of the speech wave. Journ. Acoust. Soc. Amer., vol. 50, pp. 637-655.

Babbitt M. (1961). Composition for Synthi. Cited by Boretz (1974). In: Vinton (Ed.) (1974), pp. 43-48.

Babbitt M. (1964). Philomel. Cited by Boretz (1974). In: Vinton (Ed.) (1974), pp. 43-48.

Backus J. (1961). Vibrations of the reed and air column in the clarinet. Journ. Acoust. Soc. Amer., vol. 33, pp. 806-809.

Backus J. (1970). The Acoustical Foundations of Music. John Murray, London.

Badings H. and de Bruyn J.W. (1957). In: Philips Technical Review, vol. 19(6), p. 191. Cited by Olson (1971).

Baird F.T. (1972). Computerised Musical Composition Aid - Information Transformation - Musical Notes to Electrical Signals. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Baker F.B. (1971). Computer-based instructional systems: A first look. Review of Educational Research, vol. 41, pp. 51-70.

Balfour R.D. (1972). Computerised Musical Composition Aid - Piano Interface Hardware. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Basmajian J.V. (1967). Muscles Alive. Williams and Williams Co., Baltimore.

Bauer-Mengelberg S. (1970). The Ford-Columbia input language. In: Brook (Ed.) (1970).

Beauchamp J.W. (1967). A computer system for time-variant harmonic analysis and synthesis of musical tones. Publication 992, Electrical Eng. Dept., University of Illinois, Urbana.

Beauchamp J.W. (1974). Electronic music: apparatus and technology. In: Vinton (Ed.) (1974), pp. 205-212.

Beauchamp K.G. (1975). Walsh Functions and their Applications. Academic Press, London.

Bellanger M.G., Daguet J.L. and Lepagnol G.P. (1974). Interpolation, extrapolation, and reduction of computation speed in digital filters. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-22, pp. 231-235.

Benade A.H. (1973). The physics of brasses. Scientific American, vol. 229(1), pp. 24-35.

Bennett W.R. (1948). Spectra of quantized signals. Bell Syst. Tech. Journ., vol. 27, pp.446-472.

Bergland G.D. (1969). A guided tour of the fast Fourier transform. IEEE Spectrum, vol. 6, pp. 41-52.

Blackman E.D. (1965). The physics of the piano. Scientific American, vol. 213(6), pp. 88-99.

Blankinship W.A. (1974). Note on computing autocorrelations. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-22, pp. 76-77.

Blesser B.A., Baeder K. and Zaorski R. (1975). A real-time digital computer for simulating audio systems. Journ. Audio Eng. Soc., vol. 23, pp. 698-707.

Bluestein L.I. and Rader C.M. (1965). Comparison of vector and autocorrelation pitch detectors. Journ. Acoust. Soc. Amer., vol. 37, pp. 751-752.

Bien-Ming C. (1931). The tone behaviour in Hagu: an experimental study. Archives Néerlandaises de Phonetique Expérimentale, Tome VI, pp. 6-45. Cited by McKinney (1965).

Bilsen F.A. and Ritsma R.J. (1970). Some parameters influencing the perceptibility of pitch. Journ. Acoust. Soc. Amer., vol. 47, pp. 469-475.

Bilsen F.A. (1973). On the influence of the number and phase of harmonics on the perceptibility of the pitch of complex signals. Acustica, vol. 28, pp. 60-65.

Bogert P.B., Healey M.J. and Tukey J.W. (1963). The quefrency alanysis of time series for echoes: pseudo-autocovariance, cross-cepstrum and saphe cracking. Proc. Symp. on Time Series Analysis, M. Rosenblatt, Ed., John Wiley and Sons, Inc., New York, pp. 209-243.

Böker-Heil N. (1972). Plotting conventional music notation. Journ. Music Theory, vol. 16, pp. 72-101.

Bowles E.A. (1970). Musicke's handmaiden: or Technology in the service of the Arts. In: Lincoln (Ed.) (1970), pp. 3-20.

Box G.E. and Jenkins G.M. (1970). Time Series Analysis Forecasting and Control. Holden-Day, San Francisco.

Brender M.P. and Brender R.F. (1967). Computer transcription and analysis of mid-thirteenth century musical notation. Journ. Music Theory, vol. 11, pp. 198-221.

Brook B.S. and Gould J. (1964). Notating music with ordinary typewriter characters. Fontes Artis Musicae, vol. 11, p. 142.

Brook B.S. (Ed.) (1970). Musicology and the Computer. City University of New York Press, New York.

Brooks F.P., Hopkins A.L., Newman P.G. and Wright W.V. (1957). An experiment in musical composition. IRE Trans. on Electronic Computers, vol. EC-6, pp. 175-182.

Brooks F.P. (1958). Correction. IRE Trans. on Electronic Computers, vol. EC-7, p. 60.

Brown E. (1959). Cited by Read (1974).

Brün H. (1964). Sonoriferous Loops. Cited by Vinton (Ed.) (1974), p. 107.

Bui S.T. (1977). An Ultrasonic Single Object Sensor as a Mobility Aid. Ph.D. Thesis, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand. (In Preparation).

Burg J.P. (1972). The relationship between maximum entropy spectra and maximum likelihood spectra. Geophysics, vol. 37, pp. 375-376.

Buxton W. (Ed.) (1977). Computer Music 1976/77: A Directory to Current Work. The Canadian Commission for UNESCO, Ottawa, Ontario.

Byrd D. (1974). A system for music printing by computer. Computers and the Humanities, vol. 8, pp. 161-172.

Cadzow J.A. and Martens H.R. (1970). Discrete-Time and Computer Control Systems. Prentice-Hall, Inc., New Jersey.

Caine H.L. and Ciamaga G. (1967). A preliminary report on the serial sound structure generator. Perspectives of New Music, vol. 6(1), pp. 114-118.

Cantor D. (1971). A computer program that accepts common musical notation. Computers and the Humanities, vol. 6, pp. 103-110.

Caprio J.R. (1975). Signal noise and period estimations and estimator performance bounds for multi-period observations. Conf. Rec., Int. Conf. Communications, vol. 1, pp. 1120-1123.

Cashin P. and Mayson M. (1969). Notes on the EAI 640 I/O System. Dept. Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Chadabe J. (1967). New approaches to analog-studio design. Perspectives of New Music, vol. 6(1), pp. 107-113.

Chowning J.M. (1971). The simulation of moving sound sources. Journ. Audio Eng. Soc., vol. 19, pp. 2-6.

Chowning J.M. (1973). The synthesis of complex audio spectra by means of frequency modulation. Journ. Audio Eng. Soc., vol. 21, pp. 526-534.

Chowning J.M., Grey J.M., Rush L. and Moorer J.A. (1974). Computer simulation of music instrument tones in reverberant environments. Report STAN-M-1, Centre for Computer Research in Music and Acoustics, Dept. of Music, Stanford University, Stanford, California.

Cochran W.T., Cooley J.W., Favin D.L., Helms H.D., Kaenel R.A., Lang W.A., Maling G.C., Nelson D.E., Rader C.M. and Welch P.D. (1967). What is the fast Fourier transform? IEEE Trans. Audio Electroacoust., vol. AU-15, pp. 45-55.

Cohen T.J. (1970). Source-depth determination using spectral, pseudoautocorrelation, and cepstral analysis. Geophys. Journ. Royal Astron. Soc., vol. 20, pp. 223-231.

Covell R.D., Holmes W.H. and Karbowiak A.E. (1971). Electronic instrument project at the University of New South Wales. In: The State of the Art of Electronic Music in Australia, University of Melbourne, pp. 58-62.

Cowan J.M. (1936). Pitch and intensity characteristics of stage speech. Archives of Speech, Supplement, Dec. 1936. Cited by McKinney (1965).

Crichton R.G. and Fallside F. (1974). Linear prediction model of speech production with applications to deaf speech training. Proc. IEEE, vol. 121, pp. 865-873.

Crowhurst N.H. (1975). Electronic Organs, Vol. 3. Howard W. Sams and Co., Inc., New York.

Cvetko D. (1967). Report of the Tenth Congress of the International Musicological Society, Ljubljana. Bärenreiter, University of Ljubljana.

Daggett N.L. (1966). A computer for Vocoder pitch extraction. Tech. Note 1966-3, Lincoln Laboratory, Lexington, Massachusetts. Cited by Gold and Rabiner (1969).

Dallin L. (1974). Techniques of Twentieth Century Composition. Wm. C. Brown Co., Iowa.

dal Molin A. (1973). The Music Reprographics System. Computational Musicology Newsletter, vol. 1, p. 8. Cited by Byrd (1974).

dal Molin A. (1977). The X-Y typewriters and their application as music input terminals for the computer. Proc. Second Annual Music Computation Conference, Part 4, Urbana, Illinois. Cited by Kassler (1977).

Davies H. (1974). Electronic music: history and development. In: Vinton (Ed.) (1974), pp. 212-216.

Davis W.F. (1972). A class of efficient convolution algorithms. Proc. Symp. Applications of Walsh Functions, pp. 318-329.

Diehl N.C. (1971). Computer-assisted instruction and instrumental music: implications for teaching and research. Journ. Research in Music. Ed., vol. 19, pp. 299-306.

Diehl N.C. and Ziegler R.H. (1973). Evaluation of a CAI program in articulation, phrasing and rhythm for intermediate instrumentalists. Council for Research in Music Ed. Bulletin, vol. 31, pp. 1-11.

Dolanský L.O. (1955). An instantaneous pitch-period indicator. Journ. Acoust. Soc. Amer., vol. 27, pp. 67-72.

Douglas A. (1957). Electrical Production of Music. Philosophical Library, New York.

Douglas A. (1973). Electronic Music Production. Sir Issac Pitman and Sons Ltd., London.

Dubnowski J.J., Schafer R.W. and Rabiner L.R. (1976). Real time digital hardware pitch detector. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 2-8.

Dudley H.W. (1939). System for the Artificial Production of Vocal or other Sounds. U.S. Patent No. 2,243,089, May 27, 1941. Application May 13, 1939.

Duerrenmatt H.R., Gould M. and La Rue J. (1970). Die notierung thematischer incipits auf "Mark-Sense Cards". Fontes Artis Musicae, vol. 17, pp. 15-23. Cited by Hofstetter (1976).

Eagleson H.V. and Eagleson O.W. (1947). Identification of musical instruments when heard directly over a public address system. Journ. Acoust. Soc. Amer., vol. 19, pp. 338-342.

Edmondson W.H. (1977). Novel frequency-analysis system for a vibrotactile speech training aid for the deaf. Electronic Circuits and Systems, vol. 1, pp. 57-63.

Ehrich R.W. and Foith J.P. (1976). Representation of random waveforms by relational trees. IEEE Trans. Computers, vol. C-25, pp. 725-736.

Elliot D.F. (1974). A class of generalised continuous orthogonal transforms. IEEE Trans. Acoust, Speech, Signal Processing, vol. ASSP-22, pp. 245-254.

Erickson R.F. (1968). Musical analysis and the computer: a report on some current approaches and the outlook for the future. Computers and the Humanities, vol. 3, pp. 87-104.

Fabre P. (1959). La glottographie électriques en haute fréquence, particularités de l'appareillage. Comptes Rendus des Séances de la Société de Biologie et de ses Filiales, vol. 153, pp. 1361-1364. Cited by Fourcin and Abberton (1971).

Faddeev D.K. and Faddeeva V.N. (1963). Computational Methods of Linear Algebra. W.H. Freeman, San Francisco.

Fannin P.C. (1975). Low frequency correlator using a bucket-brigade analogue delay line. Proc. IEE, vol. 122, pp. 1363-1366.

Fano R.M. (1950). Short-time autocorrelation functions and power spectra. Journ. Acoust. Soc. Amer., vol. 22, pp. 546-550.

Farnsworth D.W. (1940). High-speed motion pictures of the human vocal cords. Bell Lab. Record, vol. 18, pp. 203-208.

Fellgett P. (1973). Ambisonic reproduction of sound. Electronics and Power, vol. 19, pp. 492-494.

Fellgett P.B. (1974). Private communication to R.H.T. Bates.

Finch J.M. (1972). An overview of computer-managed instruction. Educational Technology, vol. 12(7), pp. 46-47.

Flanagan J.L. and Golden R.M. (1966). Phase vocoder. Bell Syst. Tech. Journ., vol. 45, pp. 1493-1509.

Flanagan J.L. (1972). Speech Analysis, Synthesis and Perception. Springer-Verlag, New York.

Fletcher H. (1929). Speech and Hearing. Macmillan, New York.

Forte A. (1966). A program for the analytic reading of scores. Journ. Music Theory, vol. 10, pp. 330-364.

Forte A. (1973). The Structure of Atonal Music. Yale University Press, New Haven.

Fourcin A.J. and Abberton E. (1971). First applications of a new laryngograph. Med. and Bio. Illust., vol. 21, pp. 172-182.

Fourcin A.J., Donovan R. and Roach P. (1971). Publication planned for Foliaphoniatrica. Cited by Fourcin and Abberton (1971).

Fredlund L.D. and Sampson J.R. (1973). An interactive graphics system for computer-assisted musical composition. Int. Journ. Man-Machine Studies, vol. 5, pp. 585-605.

Freedman M.D. (1967). Analysis of musical instrument tones. Journ. Acoust. Soc. Amer., vol. 41, pp. 793-806.

Freeny S.L. (1975). Special-purpose hardware for digital filtering. Proc. IEEE, vol. 63, pp. 633-648.

Frykberg S.D. and Bates R.H.T. (1977). Computers and composers. Submitted to: NUMUS WEST.

Fuller R. (1970). Toward a theory of Webernian harmony, via analysis with a digital computer. In: Lincoln (Ed.) (1970), pp. 223-276.

Gabura A.J. (1967). Musical plotter knows the score at University of Toronto. CalComp Newsletter, May/June. Cited by Byrd (1974).

Gamble W. (1923). Music Engraving and Printing. Da Capo Press, New York.

Gardner M.B. (1962). Binaural detection of single-frequency signals in presence of noise. Journ. Acoust. Soc. Amer., vol. 34, pp. 1824-1830.

Gardner M.B. (1967). Comparison of lateral localisation and distance for single- and multiple-source speech signals. Journ. Acoust. Soc. Amer., vol. 41, p. 1592, Abstract.

Gardner M.B. (1969). Image fusion, broadening and displacement in sound location. Journ. Acoust. Soc. Amer., vol. 46, pp. 339-349.

Geadah Y.A. and Corinthios M.J.G. (1977). Natural, dyadic, and sequency order algorithms and processors for the Walsh-Hadamard transform. IEEE Trans. Computers, vol. C-26, pp. 435-442.

Gethöffer H. (1973). Algebraic theory of generalised convolution on cyclic groups. Theory and Applications of Walsh and other non-sinusoidal functions, Paper 4. Hatfield, Herts, U.K. June 1973.

Ghent E. (1967a). The coordinome in relation to electronic music. Electronic Music Review, vol. 1(1), pp. 33-43.

Ghent E. (1967b). Programmed signals to performers: a new compositional resource. Perspectives of New Music, vol. 6(1), pp. 96-106.

Gibbs J.E. (1967). Walsh Spectrometry, a form of Spectral Analysis well suited to Binary Digital Computation. National Phys. Lab. Rept., Teddington, Middlesex, England.

Gibbs J.E. and Pichler F.R. (1971). Comments on transformation of "Fourier" power spectra into "Walsh" power spectra. Proc. Symp. Applications of Walsh Functions, pp. 51-54.

Gill J.S. (1961). Automatic extraction of the excitation function of speech with particular reference to the use of correlation methods. Proc. Third Int. Congress Acoust., vol. 1, Elsevier, Amsterdam, pp. 217-220.

Gold B. (1962a). Computer program for pitch extraction. Journ. Acoust. Soc. Amer., vol. 34, pp. 916-921.

Gold B. (1962b). Description of a computer program for pitch detection. Proc. Fourth Int. Congress on Acoustics, paper G 34, Harlang and Toksvig, Copenhagen.

Gold B. (1964) Note on buzz-hiss detection. Journ. Acoust. Soc. Amer., vol. 36, pp. 1659-1661.

Gold B. and Rabiner L.R. (1969). Parallel processing techniques for estimating pitch periods of speech in the time domain. Journ. Acoust. Soc. Amer., vol. 46, pp. 442-448.

Gold B., Oppenheim A.V. and Rader C.M. (1970). Theory and implementation of the discrete Hilbert transform. Proc. Symp. Comput. Process Commun., Polytechnic Press, Brooklyn, New York, pp. 235-250.

Goldstein M.H. (1957). Neurophysiological Representation of Complex Auditory Stimuli. Tech. Rept. 323, MIT Research Laboratory of Electronics, Massachusetts Institute of Technology.

Good I.J. (1971). The relation between two fast Fourier transforms. IEEE Trans. Comput., vol. C-20, pp. 310-317.

Grogono P. (1973). MUSYS: software for an electronic music studio. Software - Practice and Experience, vol. 3, pp. 369-383.

Gross R. and Leibig B. (1976). A compositionally oriented sound synthesis system. Report of the Centre for Music Experiment and Related Research, University of California at San Diego, La Jolla, California.

Gould M.J. and Logemann G.W. (1970). ALMA: alphanumeric language for music analysis. In: Brook (Ed.) (1970), pp. 57-90.

Grey J.M. (1975). An Exploration of Musical Timbre. Doctoral Thesis, Department of Psychology, Stanford University, Stanford.

Grützmacher M. and Lottermoser W. (1937). Über ein Verfahren zur trägheitsfreien Aufzeichnung von Melodiekurven. Akust. Z., vol. 2, pp. 242-248. Cited by Tove, Norman, Isaksson and Czekajewski (1966).

Harmuth H.F. (1972). Transmission of Information by Orthogonal Functions. Springer-Verlag, New York.

Harmuth H.F. (1977). Sequency Theory - Foundations and Applications. Academic Press, New York.

Harris C.M. and Weiss M.R. (1963). Pitch extraction by computer processing of high-resolution Fourier analysis data. Journ. Acoust. Soc. Amer., vol. 35, pp. 339-343.

Haskew J.R., Kelly J.M., Kelly R.M. and McKinney T.H. (1973). Results of a study of the linear prediction vocoder. IEEE Trans. Commun., vol. COM-21, pp. 1008-1014.

Hassab J.C. (1974). Time delay processing near the ocean surface. Journ. Sound and Vibration, vol. 35, pp. 489-501.

Hassab J.C. and Boucher R. (1976). A probabilistic analysis of time delay extraction by the cepstrum in stationary Gaussian noise. IEEE Trans. Information Theory, vol. IT-22, pp. 444-454.

Heckman H. (Ed.) (1967). Elektronische Datenverarbeitung in der Musikwissenschaft. Gustav Bösse Verlag, Regensburg.

Heighway J. (1976). Surface acoustic wave devices and applications. Systems Technology, No. 23, pp. 2-6.

Hiller L.A. (1959). Computer music. Scientific American, vol. 201(6), pp. 109-120.

Hiller L.A. and Isaacson L.M. (1959). Experimantal Music: Composition with an Electronic Computer. McGraw-Hill, New York.

Hiller L.A. and Baker R.A. (1964). Computer Cantata: a study in compositional method. Perspectives of New Music, vol. 3(1), pp. 62-90.

Hiller L.A. and Baker R.A. (1965). Automated music printing. Journ. Music Theory, vol. 9, pp. 129-150.

Hiller L.A. (1965). An integrated electronic music console. Journ. Audio. Eng. Soc., vol. 13, pp. 142-150.

Hiller L.A. (1970). Music composed with computers - a historical survey. In: Lincoln (Ed.) (1970), pp. 42-96.

Hofstetter F.T. (1975). GUIDO: An interactive computer-based system for improvement of instruction and research in ear-training. Journ. of Computer-Based Instruction, vol. 1, pp. 100-106.

Hofstetter F.T. (1976). Foundation, organisation, and purpose of the National Consortium for Computer-Based Musical Instruction. Music Educators National Conf., Atlantic City, New Jersey.

Hosking T.E. (1972). Computerised Musical Composition Aid - System Control. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Houtsma A.J.M. and Goldstein J.L. (1972). The central origin of the pitch of complex tones: Evidence from musical interval recognition. Journ. Acoust. Soc. Amer., vol. 51, pp. 520-529.

Hovey C.A. and Seamans D.A. (1975). A polyphonic keyboard for a voltage-controlled music synthesizer. Journ. Audio. Eng. Soc., vol. 23, pp. 459-465.

Howarth R.J. (1975). Waveform Specification and Generation. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Howarth R.J. (1977a). Automated Music Printing. M.E. Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Howarth R.J. (1977b). GUTENBURG Users Manual. Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Howe H.S. (1972). Composition limitations of electronic music synthesizers. Perspectives of New Music, vol. 10(2), pp. 120-129.

Howe H.S. (1975). Electronic Music Synthesis. J.M. Dent and Sons Ltd., London.

Hunt F.V. (1935). A direct-reading frequency meter suitable for high speed recording. The Review of Scientific Instruments, vol. 6, Feb. 1935, pp. 43-46.

Hutchins B.A. (1973). Experimental electronic music devices employing Walsh functions. Journ. Audio. Eng. Soc., vol. 21, pp. 640-645.

Hutchins B.A. (1975). Application of a real-time Hadamard transform network to sound synthesis. Journ. Audio Eng. Soc., vol. 23, pp. 558-562.

Insam E. (1973). No ladder DAC. Electronics, Dec. 20, p. 113.

Insam E. (1974). Walsh functions in waveform synthesizers. Journ. Audio Eng. Soc., vol. 22, pp. 422-425.

Itakura I. (1975). Minimum prediction residual principle applied to speech recognition. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 67-72.

Jackson R. (1967). The computer as a student of harmony. In: Cvetko (Ed.) (1967), pp. 435-450.

Johnson D.A.H. (1975). Number Theory Transforms for Fast Convolution. Report 3-1975, Dept. of Electrical Engineering, University of Melbourne, Parkville, Australia.

Jolley L.B.W. (1961). Summation of Series. Dover, New York.

Jordan R.B. (1974). Computer Modelling of Adrenal Function. Ph.D. Thesis, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Kassler M. (1970). MIR - a simple programming language for musical information retrieval. In: Lincoln (Ed.) (1970), pp. 299-327.

Kassler M. and Howe H.S. (1975). Computers and Music. To appear in: Grove's Dictionary of Music and Musicians, 6th Ed., Macmillan, London.

Kassler M. (1977). Computer-Assisted Music Printing. Report to the Music Board of the Australia Council.

Keeler J.S. (1972). Piecewise-periodic analysis of almost-periodic sounds and musical transients. IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 338-344.

Kennett B.L.N. (1970). A note on the finite Walsh transform. IEEE Trans. Information Theory, vol. IT-16, pp. 489-491.

Kemerait R.C. and Childers D.G. (1972). Signal detection and extraction by cepstrum techniques. IEEE Trans. Information Theory, vol. IT-18, pp. 745-759.

Kindlmann P.J. and Fuge P.H. (1968). Sound synthesis: a flexible modular approach with integrated circuits. IEEE Trans. Audio Electroacoust., vol. AU-16, pp. 507-514.

Knowlton P.H. (1971). Interactive Communication and Display of Keyboard Music. Ph.D. dissertation, University of Utah, Salt Lake City.

Knowlton P.H. (1972). Capture and display of keyboard music. Datamation, pp. 56-60.

Knudsen M.J. (1975). Real-time linear-predictive coding of speech on the SPS-41 triple-microprocessor machine. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 140-145.

Knuth D.E. (1969). The Art of Computer Programming, vol. 2, Semi-numerical Algorithms. Addison-Wesley, Reading, Massachusetts.

Knuth D.E. (1971). The art of computing programming - errata et addenda. Rep. STAN-CS-71-194, Computer Science Dept., Stanford University, Stanford, California, pp. 21-26. Cited by Agarwal and Burrus (1974).

Kopec G.E., Oppenheim A.V. and Tribolet J.M. (1977). Speech analysis by homomorphic prediction. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 40-49.

Kuhn W. (1974). Computer-assisted instruction in music: Drill and practice in ear-training. College Music Symposium, vol. 14, pp. 89-101.

Lamb M.R. (1977). The Computer as a Musicianship Teaching Aid. Ph.D. Thesis, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand. (In Preparation).

Landwehr G. (1973). A comparison of discrete Walsh and Fourier transforms. Theory and Applications of Walsh and other non-sinusoidal functions, Paper 9. Hatfield, Herts, U.K. June 1973.

Lange F.H. (1967). Correlation Techniques. Iliffe Books, London.

Laske O.E. (1973). Introduction to a Generative Theory of Music. Sonological Report No. 1, Institute of Sonology, Utrecht State University, Utrecht.

Laske O.E. (1974). Towards a musical intelligence system: OBSERVER. NUMUS WEST, No. 5, pp. 11-17.

Lathi B.P. (1965). Signals, Systems and Communication. John Wiley and Sons, Inc., New York.

Lefkoff G. (1967a). Computers and the study of musical style. In: Lefkoff G. (Ed.) (1967b), pp. 43-61.

Lefkoff G. (Ed.) (1967b). Computer Applications in Music. West Virginia University Library, Morgantown.

Lefkoff G. (1970). Automated discovery of similar segments in the forty-eight permutations of a twelve-tone row. In: Lincoln (Ed.) (1970), pp. 147-153.

Lehman P.R. (1964). Harmonic structure of the tone of the bassoon. Journ. Acoust. Soc. Amer., vol. 36, pp. 1649-1653.

Lerner R.M. (1959). A Method of Speech Compression. Sc.D. thesis, Massachusetts Institute of Technology.

Lewer S.K. (1948). Electronic Musical Instruments. Electronic Engineering, London.

Licklider J.C.R. (1946). Effects of amplitude distortion upon the intelligibility of speech. Journ. Acoust. Soc. Amer., vol. 18, pp. 429-434.

Licklider J.C.R. and Pollack I. (1948). Effects of differentiation, integration, and infinite peak clipping upon the intelligibility of speech. Journ. Acoust. Soc. Amer., vol. 20, pp. 42-51.

Licklider J.C.R. (1951). A duplex theory of pitch perception. Experimentia, vol. 7, p. 128. Cited by Schouten, Ritsma and Cardozo (1962).

Licklider J.C.R. (1954). Periodicity pitch and place pitch. Journ. Acoust. Soc. Amer., vol. 26, p. 945, Abstract.

Licklider J.C.R. (1956). Auditory frequency analysis. Proc. 3rd London Symposium on Information Theory, C. Cherry (Ed.), Butterworths Scientific Publications, London, pp. 253-268. Cited by Schouten, Ritsma and Cardozo (1962).

Lincoln H.B. (Ed.) (1970). The Computer and Music. Ithaca, New York.

Longuet-Higgins H.C. and Steedman M.J. (1971). On interpreting Bach. Machine Intelligence, vol. 6, pp. 221-241.

Longuet-Higgins H.C. (1976). Perception of melodies. Nature, vol. 263, pp. 646-653.

Lopresti P.V. and Suri H.L. (1974). A fast algorithm for the estimation of autocorrelation functions. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-22, pp. 449-453.

Lovelock W. (1946). First Year Harmony. A. Hammond and Co., London.

Lincoln H.B. (1970). The Computer and Music. Ithaca, New York.

Luce D.A. (1963). Physical Correlates of Nonpercussive Musical Instrument Tones. Ph.D. thesis, Massachusetts Institute of Technology.

Luce D. and Clark M. (1965). Durations of attack transients on nonpercussive orchestral instruments. Journ. Audio Eng. Soc., vol. 13, pp. 194-199.

Luce D. and Clark M. (1967). Physical correlates of brass instrument tones. Journ. Acoust. Soc. Amer., vol. 42, pp. 1232-1243.

Lui K.Y., Reed I.S. and Truong T.K. (1976). Fast number-theoretic transforms for digital filtering. Electron. Letts., vol. 12, pp. 644-646.

Makhoul J. (1973). Spectral analysis of speech by linear prediction. IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 140-148.

Makhoul J. (1975). Linear prediction: a tutorial review. Proc. IEEE, vol. 63, pp. 561-580.

Maksym J.N. (1972). Iterative Adjustment of Predictive Quantizers. Ph.D. dissertation, Department of Electrical Engineering, Carleton University, Ottawa, Ontario.

Maksym J.N. (1973). Real-time pitch extraction by adaptive prediction of the speech waveform. IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 149-154.

Markel J.D. (1971). FFT pruning. IEEE Trans. Audio Electroacoust., vol. AU-19, pp. 305-311.

Markel J.D. (1972a). Digital inverse filtering - a new tool for formant trajectory estimation. IEEE Trans. Audio Electroacoustic., vol. AU-20, pp. 129-137.

Markel J.D. (1972b). The SIFT algorithm for fundamental frequency estimation. IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 367-377.

Markel J.D. (1973). Application of a digital inverse filter for automatic formant and F_o analysis. IEEE Trans. Audio Electroacoust., vol. AU-21, pp. 154-160.

Markel J.D. and Gray A.H. (1974). A linear prediction vocoder simulation based upon the autocorrelation method. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-22, pp. 124-134.

Mars P. and Cattanach J.M. (1977). Automatic transcription of keyboard music. Proc. IEE, vol. 124, pp. 436-440.

Martin D.W. (1942). Lip vibrations in a cornet mouthpiece. Journ. Acoust. Soc. Amer., vol. 13, pp. 305-308.

Mathews M.V. (1961). An acoustic compiler for music and psychological stimuli. Bell Syst. Tech. Journ., vol. 40, pp. 677-694.

Mathews M.V., Miller J.E. and David E.E. (1961). Pitch synchronous analysis of voiced sounds. Journ. Acoust. Soc. Amer., vol. 33, pp. 179-186.

Mathews M.V. (1963). The digital computer as a musical instrument. Science, vol. 142, pp. 553-557.

Mathews M.V. (1966). A graphical language for composing and playing sounds and music. Presented at 31st Convention of Audio Eng. Soc., October 1966.

Mathews M.V. (1969). The Technology of Computer Music. The M.I.T. Press, Cambridge, Massachusetts.

Mathews M.V. and Moore F.R. (1970a). Groove - a program to compose, store, and edit functions of time. Commun. A.C.M., vol. 13, pp. 715-721.

Mathews M.V. and Moore F.R. (1970b). GRooVE - a computer program for real-time music and sound synthesis. Journ. Acoust. Soc. Amer., vol. 47, p. 132, Abstract.

Mathews M.V., Moore F.R. and Risset J.C. (1974). Computers and future music. Science, vol. 183, pp. 263-268.

Mayson M.R. (1971). A Storage Tube Computer Display System. Memo 83, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

McClellan J.H. (1976). Hardware realisation of a Fermat number transform. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 216-225.

McGonegal C.A., Rabiner L.R. and Rosenberg A.E. (1975). A semi-automatic pitch detector (SAPD). IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 570-574.

McKinney N.P. (1965). Laryngeal Frequency Analysis for Linguistic Research. Communication Sciences Laboratory, Report 14, University of Michigan.

Meertens L. (1968). Quartet No.1 in C Major. Cited by Kassler M. (1968): Report from Edinburgh. Perspectives of New Music, vol. 7(2), pp. 175-177.

Mendel A. (1969). Some preliminary attempts at computer-assisted style analysis in music. Computers and the Humanities, vol. 4, pp. 41-52.

Metfessel M. (1926). Technique for objective studies of the vocal art. University of Iowa Studies in Psychology, vol. 9, pp. 1-40.

Meyer E.A. and Schneider C. (1913). Theorie des Tonhöhen-Messapparates. Vox, vol. 23, no. 3, pp. 152-163. Cited by McKinney (1965).

Miller R.L. (1953). U.S. Patent 2,627,541. Feb. 1953. Cited by Schroeder (1968).

Miller R.L. and Weibel E.S. (1956). Measurement of the fundamental period of speech using a delay line. Presented at the 51st Meeting of the Acoustical Society of America.

Miller R.L. (1959). Nature of the vocal cord wave. Journ. Acoust. Soc. Amer., vol. 31, pp. 667-677.

Miller N.J. (1975). Pitch detection by data reduction. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 72-79.

Mitra S.K. (1971). Active Inductorless Filters. IEEE Press, New York.

Moog R.A. (1965). Voltage-controlled electronic music modules. Journ. Audio Eng. Soc., vol. 13, pp. 200-206.

Moog R.A. (1967). Electronic music: its composition and performance. Electronics World, February, pp. 42-46.

Moorer J.A. (1974). The optimum comb method of pitch period analysis of continuous digitized speech. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-22, pp. 330-338.

Moorer J.A. (1975). On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer. Ph.D. dissertation, Dept. of Computer Science, Stanford University, Stanford.

Moorer J.A. (1976). The synthesis of complex audio spectra by means of discrete summation formulas. Journ. Audio Eng. Soc., vol. 24, pp. 717-727.

Moorer J.A. (1977). Signal processing aspects of computer music: a survey. Proc. IEEE, vol. 65, pp. 1108-1137.

Morris L.R. (1977). Automatic generation of time efficient digital signal processing software. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 74-79.

Mukwamataba J. (1972). Computerised Musical Composition Aid - System Studies. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Nakamura S. (1976). A digital correlator using delta modulation. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 238-243.

Newell A., Shaw J.C. and Simon H.A. (1958). Report on a general problem-solving program. The Rand Corporation, P-1584. Cited by Lincoln (Ed.) (1970), p. 72.

Newall A., Shaw J.C. and Simon H.A. (1960). Report on a general problem-solving program for a computer. Information processing: Proc. International Conf. on Information Processing, pp. 256-264. UNESCO, Paris.

Nicholson P.J. (1971). Algebraic theory of finite Fourier transforms. Journ. Comput. Syst. Sci., vol. 5, pp. 524-547. Cited by Agarwal and Burrus (1974).

Noll A.M. (1964). Short-time spectrum and "cepstrum" techniques for vocal-pitch detection. Journ. Acoust. Soc. Amer., vol. 36, pp. 296-302.

Noll A.M. (1967). Cepstrum pitch determination. Journ. Acoust. Soc. Amer., vol. 41, pp. 293-309.

Noll A.M. (1968). Clipstrum pitch determination. Journ. Acoust. Soc. Amer., vol. 44, pp. 1585-1591.

Noll A.M. (1970). Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum and a maximum likelihood estimate. MRI Symp. Proc., vol. 19, Polytechnic Press, Brooklyn, New York, pp. 779-797.

Nussbaumer H. (1977). Digital filtering using pseudo Fermat number transforms. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 79-83.

Obata J. and Kobayashi R. (1937). A direct-reading pitch recorder and its applications to music and speech. Journ. Acoust. Soc. Amer., vol. 9, pp. 156-161.

Obata J. and Kobayashi R. (1938). An apparatus for direct-recording the pitch and intensity of sound. Journ. Acoust. Soc. Amer., vol. 10, pp. 147-149.

Oetken G., Parks T.W. and Schüssler H.W. (1975). New results in the design of digital interpolators. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 301-309.

Ohnsorg F.R. (1971). Spectral modes of the Walsh-Hadamard transform. Proc. Symp. Applications of Walsh Functions, pp. 55-59.

Olson H.F. and Belar H. (1955). Electronic music synthesizer. Journ. Acoust. Soc. Amer., vol. 27, pp. 595-612.

Olson H.F. and Belar H. (1961). Aid to music composition employing a random probability system. Journ. Acoust. Soc. Amer., vol. 23, p. 1163.

Olson H.F. (1971). Electronic music synthesis for recordings. IEEE Spectrum, vol. 8(4), pp. 18-29.

Oppenheim A.V., Schafer R.W. and Stockham T.G. (1968). Non-linear filtering of multiplied and convolved signals. Proc. IEEE, vol. 56, pp. 1264-1291.

Oppenheim A.V. (1969). Speech analysis-synthesis system based on homomorphic filtering. Journ. Acoust. Soc. Amer., vol. 45, pp. 458-465.

Oppenheim A.V. and Schafer R. (1975). Digital Signal Processing. Prentice-Hall, New Jersey.

Page E.S. and Wilson L.B. (1973). Information Representation and Manipulation in a Computer. Cambridge University Press, London.

Papoulis A. (1962). The Fourier Integral and its Applications. McGraw-Hill Book Co., Inc., New York.

Patrick P.H. (1974). A computer study of a suspension-formation in the masses of Josquin des Prez. Computers and the Humanities, vol. 8, pp. 321-331.

Peled A. (1976). On the hardware implementation of digital signal processors. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 76-86.

Peters G.D. (1974). Feasibility of Computer-Assisted Instruction for Instrumental Music Education. Ed.D. dissertation, University of Illinois. Cited by Hofstetter (1976).

Peters G.D. (1975). The Development of Computer-Assisted Instructional Materials for Use in the Teaching of Instrumental Music via PLATO IV. Final Report, Undergraduate Instructional Award, University of Illinois. Cited by Hofsetter (1976).

Peterson T.L. (1975). Vocal tract modulation of instrumental sounds by digital filtering. Presented at the Music Computation Conf. II, School of Music, University of Illinois, Urbana-Champaign, Nov. 7-9, 1975. Cited by Moorer (1977).

Petersen T.L. (1976a). Dynamic sound processing. In: Proc. 1976 ACM Computer Science Conf., Anaheim, California. Cited by Moorer (1976).

Petersen T.L. (1976b). Analysis-synthesis as a tool for creating new families of sound. Presented at 54th Conv. Audio Eng. Soc., Los Angeles, California, May 4-7, 1976. Cited by Moorer (1976).

Pichler F.R. (1970). Technical Research Rept. R-70-11, Dept. of Electrical Engineering, University of Maryland. Cited by Gibbs and Pichler (1971).

Pierce J.R. (1961). Symbols, Signals and Noise. Harper, New York, pp. 250-261. Cited by Lincoln (Ed.) (1970), p. 14.

Pinkerton R.C. (1956). Information theory and melody. Scientific American, vol. 194(2), pp. 77-86.

Piston W. (1947). Counterpoint. Norton, New York.

Pitassi D.A. (1971). Fast convolution using the Walsh transform. Proc. Symp. Applications of Walsh Functions, pp. 130-133.

Placek R.W. (1974). Design and trial of a computer-assisted lesson in rhythm. Journ. of Research in Music Ed., vol. 22, pp. 13-23.

Plomp R. and Smoorenburg G.F. (Eds) (1970). Frequency Analysis and Periodicity Detection in Hearing. A.W. Sythoff, Leiden.

Pollack I. (1952). The information of elementary auditory displays. Journ. Acoust. Soc. Amer., vol. 24, pp. 745-749.

Pollack I. and Ficks L. (1954). Information of elementary multidimensional auditory displays. Journ. Acoust. Soc. Amer., vol. 26, pp. 154-158.

Pollard J.M. (1971). The fast Fourier transform in a finite field. Math Comput., vol. 25, pp. 365-374.

Prerau D.S. (1970). Computer Pattern Recognition of Printed Music. Ph.D. thesis, Electrical Engineering Dept., Massachusetts Institute of Technology.

Prerau D.S. (1971). Computer pattern recognition of printed music. AFIPS Fall Joint Computer Conference, pp. 153-162.

Prieberg F.K. (1960). Musica ex Machina. Ullstein, Berlin.

Pruslin D.H. (1967). Automatic Recognition of Sheet Music. Ph.D. thesis, Electrical Engineering Dept., Massachusetts Institute of Technology. Cited by Prerau (1971).

Pulfer J.K. (1970). Man-machine interaction in creative applications. Int. Journ. Man-Machine Studies, vol. 3, pp. 1-11.

Rabiner L.R. and Rader C.M. (1972). Digital Signal Processing. IREE Press, New York.

Rabiner L.R. and Gold B. (1975). Theory and Application of Digital Signal Processing. Prentice-Hall, New Jersey.

Rabiner L.R., Sambur M.R. and Schmidt C.E. (1975). Applications of a non-linear smoothing algorithm to speech processing. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-23, pp. 552-557.

Rabiner L.R., Cheng M.J., Rosenberg A.E. and McGonegal C.A. (1976). A comparative performance study of several pitch detection algorithms. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 399-418.

Rabiner L.R. and Schafer R.W. (1976). Digital techniques for computer voice response: implementations and applications. Proc. IEEE, vol. 64, pp. 416-433.

Rabiner L.R. (1977). On the use of autocorrelation analysis for pitch detection. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 24-33.

Rader C.M. (1964). Vector pitch detection. Journ. Acoust. Soc. Amer., vol. 36, p. 1963.

Rader C.M. (1968). Discrete Fourier transforms when the number of data samples is prime. Proc. IEEE, vol. 56, pp. 1107-1108.

Rader C.M. (1972a). The number theoretic DFT and exact discrete convolution. IEEE Arden House Workshop on Digital Signal Processing, Harriman, New York.

Rader C.M. (1972b). Discrete convolution via Mersenne transforms. IEEE Trans. Comput., vol. C-21, pp. 1269-1273.

Rader C.M. and Brenner N.M. (1976). A new principle for fast Fourier transformation. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 264-266.

Railsback O.L. (1937). A chromatic stroboscope. Journ. Acoust. Soc. Amer., vol. 9, pp. 37-42.

Raven-Hart R. (1930). The Martenot instrument. Wireless World, vol. 27, p. 58.

Raskin J. (1967). A Hardware Independent Computer Graphics System. Masters Thesis, Dept. of Computer Science, Pennsylvania State University.

Read G. (1974). Music Notation - A Manual of Modern Practice. Victor Gollancz Ltd, London.

Reddy D.R. (1967). Pitch period determination of speech sounds. Comm. ACM, vol. 10, pp. 343-348.

Reed I.S. and Truong T.K. (1975). The use of finite fields to compute convolutions. IEEE Trans. Information Theory, vol. IT-21, pp. 208-213.

Reed I.S. and Truong T.K. (1976). Convolutions over residue classes of quadratic integers. IEEE Trans. Information Theory, vol. IT-22, pp. 468-745.

Regener E. (1967). A multiple pass transcription and a system for music analysis by computer. In: Heckman (Ed.) (1967), pp. 89-102.

Reitman W.R. (1960). Information processing languages and heuristic programming. Bionics Symposium (WADD Tech. Report 60-600), Wright-Patterson Air Force Base, Ohio, Directorate of Advanced Systems Technology. Cited by Lincoln (Ed.) (1970), p. 72.

Reitboeck H. and Brody T.P. (1968). A transformation with invariance under cyclic permutation for applications in pattern recognition. Scientific Paper 68-1F1-ADAPT-P1, Westinghouse Research Labs. Cited by Ulman (1970).

Richardson E.G. (1954). The transient tones of wind instruments. Journ. Acoust. Soc. Amer., vol. 26, pp. 960-962.

Risset J.C. (1970). An Introductory Catalog of Computer Synthesized Sounds. Bell Telephone Laboratories, Murray Hill, New Jersey.

Ritchie G.R. and Turner J.A. (1975). Input devices for interactive graphics. Int. Journ. Man-Machine Studies, vol. 7, pp. 639-660.

Roberts A. (1966). An all-FORTRAN music-generating computer program. Journ. Audio Eng. Soc., vol. 14, pp. 17-20.

Robinson T.D. (1967). IML-MIR: a data-processing system for the analysis of music. In: Heckman (Ed.) (1967), pp. 103-135.

Robinson G.S. (1972). Logical convolution and discrete Walsh and Fourier power spectra. IEEE Trans. Audio Electroacoust., vol. AU-20, pp. 271-280.

Robson A.B. (1976). Private communication.

Roche D.J. (1972). Computerised Musical Composition Aid - Display Software. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Rogers J.A.V. (1976). GASP: a programmable signal processor. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Philadelphia.

Roller G. (1965). Development of a method for analysis of musical compositions using an electronic digital computer. Journ. Research in Music Education, vol. 13, pp. 249-252.

Rom R. (1975). On the cepstrum of two-dimensional functions. IEEE Trans. Information Theory, vol. IT-21, pp. 214-217.

Rosenblith W.A. and Stevens K.N. (1953). On the DL for frequency. Journ. Acoust. Soc. Amer., vol. 25, pp. 980-985.

Rosenboom D. (Ed.) (1976). Biofeedback and the Arts: Results of Early Experiments. Aesthetic Research Centre of Canada.

Ross T. (1970). The Art of Music Engraving and Processing. Hansen Press, Miami.

Ross M.J., Shaffer H.L., Cohen A., Freudberg R. and Manley H.J. (1974). Average magnitude difference function pitch extractor. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-22, pp. 353-362.

Rubin A.J., Keener D.H. and Downer S.W. (1975). A future for hybrids? Research/Development, vol. 26, no. 5, pp. 38-44.

Sambur M.R. (1975). An efficient linear prediction vocoder. Bell System Technical Journ., vol. 54(10), pp. 1693-1723.

Sambur M.R. and Jayant N.S. (1976). Speech encryption by manipulation of LPC parameters. Bell System Technical Journ., vol. 55, pp. 1373-1388.

Sandlund B. (1972). Presentation of the Soundprocessor Developed at EMS. EMS Information No. 4, Stiftelsen Electronmusikstudion, Stockholm.

Schafer R.W. and Rabiner L.R. (1970). System for automatic formant analysis of voiced speech. Journ. Acoust. Soc. Amer., vol. 47, pp. 634-648.

Schoeff J.A. and Reid J.S. (1976). How Telco PCM benefits from new IC technology. Telephone Engineer and Management, vol. 80(7), pp. 29-33.

Schonhage A. and Strassen V. (1971). Fast multiplication of large numbers. (In German). Comput., vol. 7, pp. 281-292. Cited by Agarwal and Burrus (1974).

Schouten J.F. (1940). The residue and the mechanism of hearing. Proc. Kon. Ned. Akad. v. Wetensch., vol. 43, p. 991. Cited by Bilsen (1973).

Schouten J.F., Ritsma R.J. and Cardozo B.L. (1962). Pitch of the residue. Journ. Acoust. Soc. Amer., vol. 34, pp. 1418-1424.

Schroeder M.R. and Atal B.S. (1962). Generalized short-time power spectra and autocorrelation functions. Journ. Acoust. Soc. Amer., vol. 34, pp. 1679-1683.

Schroeder M.R. (1966). Vocoders: analysis and synthesis of speech. Proc. IEEE, vol. 54, pp. 720-734.

Schroeder M.R., Flanagan J.L. and Lundry E.A. (1967). Bandwidth compression of speech by analytic-signal rooting. Proc. IEEE, vol. 55, pp. 396-401.

Schroeder M.R. (1968). Period histogram and product spectrum: new methods for fundamental-frequency measurement. Journ. Acoust. Soc. Amer., vol. 43, pp. 829-834.

Schroeder M.R. (1970). Parameter estimation in speech: a lesson in unorthodoxy. Proc. IEEE, vol. 58, pp. 707-712.

Scott N.G. (1975). Equipment for electronic music synthesisers. Presented at National Electronics Conference (NELCON), Wellington, New Zealand, August 1975.

Scripture E.W. (1902). The Elements of Experimental Phonetics. Charles Scribner's Sons, New York. Cited by McKinney (1965).

Scripture E.W. (1903). A record of the melody of the Lord's Prayer. Die Neueren Sprachen, Heft 9, Band 10. Cited by McKinney (1965).

Scripture E.W. (1923). The study of English speech by new methods of phonetic investigation. Proc. Brit. Academy, vol. 10, pp. 270-299. Cited by McKinney (1965).

Seashore C.E. (1932). The vibrato. University of Iowa Studies in the Psychology of Music, Vol. 1.

Seeger C. (1951). An instantaneous music notator. Journ. Int. Folk Music Council, vol. 3, p. 103.

Seeger C. (1957). Toward a universal music sound-writing for musicology. Journ. Int. Folk Music Council, vol. 9, p. 63.

Shanks J.L. (1967). Recursion filters for digital processing. Geophysics, vol. 32(1), pp. 33-51.

Shanks J.L. (1969). Computation of the fast Walsh-Fourier transform. IEEE Trans. Computers, vol. C-18, pp. 457-459.

Shannon C.E. and Weaver W. (1959). Mathematical Theory of Communication. University of Illinois Press, Urbana.

Shepard R.N. (1964). Circularity in judgements of relative pitch. Journ. Acoust. Soc. Amer., vol. 36, pp. 2346-2353.

Shower E.G. and Biddulp R. (1932). Differential pitch sensitivity of the ear. Journ. Acoust. Soc. Amer., vol. 3, pp. 275-287.

Silverman H.F. (1977). An introduction to programming the Winograd Fourier Transform Algorithm (WFTA). IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-25, pp. 152-165.

Singleton R.C. (1969). An algorithm for computing the mixed radix fast Fourier transform. IEEE Trans. Audio Electroacoust., vol. AU-17, pp. 93-103.

Skinner D.P. (1976). Pruning the decimation in-time FFT algorithm. IEEE Trans., Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 193-194.

Slawson W. (1967). Wishful Thinking About Winter. Cited by Dallin (1974), p. 259.

Slawson W. (1968). Movements for Orchestra and Tape. Cited by Slawson (1969), p. 122.

Slawson W. (1969). A speech-oriented synthesiser of computer music. Journ. Music Theory, vol. 13, pp. 94-127.

Smith L. (1973). Editing and printing music by computer. Journ. Music Theory, vol. 17, pp. 292-309.

Smoliar S.W. (1972). Music theory - a programming linguistic approach. Proc. ACM Annual Conf., pp. 1001-1014.

Smoliar S.W. (1973a). Basic research in computer-music studies. Interface, vol. 2, pp. 121-125.

Smoliar S.W. (1973b). A data structure for an interactive music system. Interface, vol. 2, pp. 127-140.

Smoliar S.W. (1974). Process structuring and music theory. Journ. Music Theory, vol. 18, pp. 308-336.

Sondhi M.M. (1964). Equivalence of "vector" and autocorrelation pitch detectors. Journ. Acoust. Soc. Amer., vol. 36, p. 1964.

Sondhi M.M. (1968). New methods of pitch extraction. IEEE Trans. Audio Electroacoust., vol. AU-16, pp. 262-266.

Sonesson B. (1960). On the anatomy and vibratory pattern of the human vocal folds with special reference to a photoelectrical method for studying vibratory movements. Acta Otolaryngologica, Suppl. vol. 156, pp. 1-80.

Stansfield E.V. and Bogner R.E. (1973). Determination of vocal-tract-area function from transfer impedance. Proc. IEE, vol. 120, pp. 153-158.

Stockham T.G. (1966). High speed convolution and correlation. AFIPS Conf. Proc., vol. 28, pp. 229-233.

Stone R.B. and White G.M. (1963). Digital correlator detects voice fundamental. Electronics, vol. 36, no. 4, pp. 28-30.

Strong W. and Clark M. (1967a). Synthesis of wind-instrument tones. Journ. Acoust. Soc. Amer., vol. 41, pp. 39-52.

Strong W. and Clark M. (1967b). Perturbations of synthetic orchestral wind-instrument tones. Journ. Acoust. Soc. Amer., vol. 41, pp. 277-285.

Styles B.C. (1974). Describing music to a computer. Int. Journ. Man-Machine Studies, vol. 6, pp. 125-134.

Suchoff B. (1968). Computerised folk song research and the problems of variants. Computers and the Humanities, vol. 2, pp. 155-158.

Sundberg T. (1977). The acoustics of the singing voice. Scientific American, vol. 236(3), pp. 82-91.

Tanaka H. (1972). Hadamard transform for speech wave analysis. Report STAN-CS-307, Computer Science Dept., Stanford University.

Taub H. and Schilling D.L. (1971). Principles of Communication Systems. McGraw Hill, New York.

Taylor C.A. (1965). The Physics of Musical Sounds. English Universities Press, London.

Taylor E. (1972). Revised draft application to the Calouste Gulbenkien Foundation for finance to develop a computer-controlled digital sound system for the production of electronic music. Faculty of Music, University of Durham.

Tenney J. (1963). Sound generation by means of a digital computer. Journ. Music Theory, vol. 7, pp. 24-70.

Terhardt E. (1972). Zür Tonhöhenwahrnehmung von Klängen. I. Psychoakustische Grundlagen. Akustica, vol. 26, pp. 173-186.

Tiffin J. (1932). Phonotographic apparatus. University of Iowa Studies in the Psychology of Music, Vol. 1, pp. 118-133.

Tove P.A., Norman B., Isaksson L. and Czekajewski J. (1966). Direct-recording frequency and amplitude meter for analysis of musical and other sonic waveforms. Journ. Acoust. Soc. Amer., vol. 39, pp. 362-371.

Truax B.D. (1973a). Some programs for real-time computer synthesis and composition. Interface, vol. 2, pp. 159-163.

Truax B.D. (1973b). The Computer Composition - Sound Synthesis Programs POD4 and POD5. Sonological Report No. 2, Institute of Sonology, Utrecht State University.

Truax B.D. (1974). General techniques of composition programming. NUMUS WEST, No. 5, pp. 17-20.

Tucker W.H. (1972). Computerised Musical Composition Aid - Software Control and Editing. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Tucker W.H. (1974). Linear prediction and its applications. Unpublished report to J.H. Andreae and R.H.T. Bates.

Tucker W.H., Lamb M.R., Howarth R.J., Vaughan R.G., Kennedy W.K., Frykberg S.D. and Bates R.H.T. (1975). Computerised musicianship aids. Presented at National Electronics Conference (NELCON), Wellington, New Zealand, August 1975.

Tucker W.H., Bates R.H.T., Frykberg S.D., Howarth R.J., Kennedy W.K., Lamb M.R. and Vaughan R.G. (1977). An interactive aid for musicians. Accepted for publication in: Int. Journ. Man-Machine Studies.

Ulman L.J. (1970). Computation of the Hadamard transform and the R-transform in ordered form. IEEE Trans. Computers, vol. C-19, pp. 359-360.

Ulrych T.J. (1971). Application of homomorphic deconvolution to seismology. Geophysics, vol. 36, pp. 650-660.

Ussachevsky V. (1958). The process of experimental music. Journ. Audio Eng. Soc., vol. 6, pp. 202-207.

Van den Bos A. (1971). Alternative interpretation of maximum entropy spectral analysis. IEEE Trans. Information Theory, vol. IT-17, pp. 493-494.

Vaughan R.G. (1975). Digital Organ Project. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Vaughan R.G. (1977). A Computer Controlled Digital Music Synthesiser. M.E. Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Vegh E. and Leibowitz L.M. (1976). Fast complex convolution in finite rings. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 343-344.

Vinton J. (Ed.) (1974). Dictionary of Twentieth Century Music. Thames and Hudson, London.

Voss R.M. (1965). Brandeis University experimental music studio. Journ. Audio Eng. Soc., vol. 13, pp. 65-68.

Wallach H., Newman E.B. and Rosenzweig M.R. (1949). The precedence effect in sound localization. Amer. Journ. Psychology, vol. 52, pp. 315-336.

Ward W.D. (1954). Subjective musical pitch. Journ. Acoust. Soc. Amer., vol. 26, pp. 369-380.

Weiss M.R., Vogel R.P. and Harris C.M. (1966). Implementation of a pitch extractor of the double-spectrum-analysis type. Journ. Acoust. Soc. Amer., vol. 40, pp. 657-662.

Wells I.S. (1972). Computerised Musical Composition Aid - Storage Software. Third Professional Project Report, Electrical Engineering Dept., University of Canterbury, Christchurch, New Zealand.

Wenker J. (1970). A computer-oriented music notation including ethnomusicological symbols. In: Brook (Ed.) (1970), pp. 91-129.

Westman H.P. (Ed.) (1968). Reference Data for Radio Engineers. Howard W. Sams and Co., Inc., New York.

Winograd T. (1968). Linguistics and the computer analysis of tonal harmony. Journ. Music Theory, vol. 12, pp. 2-49.

Winograd S. (1976). On computing the discrete Fourier transform. IBM Research Report RC-6291.

Wise J.D., Caprio J.R. and Parks T.W. (1976). Maximum likelihood pitch estimation. IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-24, pp. 418-423.

Witfield I.C. (1957). The physiology of hearing. Progress in Biophys. and Biophys. Chem., vol. 8, p. 43. Cited by Schouten, Ritsma and Cardozo (1962).

Witten I.H. (1977). Personal Communication.

Wuorinen C. (1970). Time's Encomium. Cited by Olsen (1971), p. 30.

Xenakis I. (1963). Musiques Formelles. Editions Richard-Masse, Paris.

Xenakis I. (1971). Formalised Music.
Indianna University Press, Bloomington.

Youngblood J. (1970). Root progression and composer identification. In: Lincoln (Ed.) (1970), pp. 172-180.

Yuen C. (1971). Walsh functions and Gray code. Proc. Symp. Applications of Walsh Functions, pp. 68-73.

Zinovieff P. (1968). A computer-controlled electronic music studio. DECUS Fourth European Seminar, Maynard, Massachusetts, pp. 139-145.